Local Readout Ground Station
LRGS-DOMSAT

User's and Administrator's Guide
Version 3.4

Revision 1.1

July 15, 2003

Ilex Engineering, Inc
5114 Crystal Park Lane
Ellicott City, MD 21043
Tel: 410.465.6948
Email: info@ilexeng.com

Table of Contents

1. The NOAA/NESDIS Data Collection System

The National Environmental Satellite, Data, and Information Service (NESDIS) operates the U.S. Geostationary Operational Environmental Satellite (GOES) system. The GOES system's primary mission is to continuously observe changing weather phenomena from satellite-based sensors situated approximately 23,000 miles from Earth. As a collateral duty, the GOES system supports a radio relay or Data Collection System (DCS). The DCS enables a large variety of environmental data to be relayed from point sources through GOES and back to Earth, from where these data are disseminated to the various system users. These point sources are called Data Collection Platforms (DCP), and can be are land, sea or mobile-based.

The DCS Automated Processing System (DAPS) is the hub of this largs data-relay system. DAPS is a large dual-computer based system located at the NOAA Command and Data Acquisition (CDA) facility in Wallops, Virginia. It continually monitors all GOES RF channels for incoming DCP messages. DAPS can support the receipt and archival of messages from up to 100,000 platforms, redistributing them to up to 5,000 users. Currently, over 12,000 platforms are active in the DCS.

DAPS supports the distribution of DCP messages to the user-organizations via a domestic communications satellite (DOMSAT). DAPS continuously broadcasts all incoming DCP

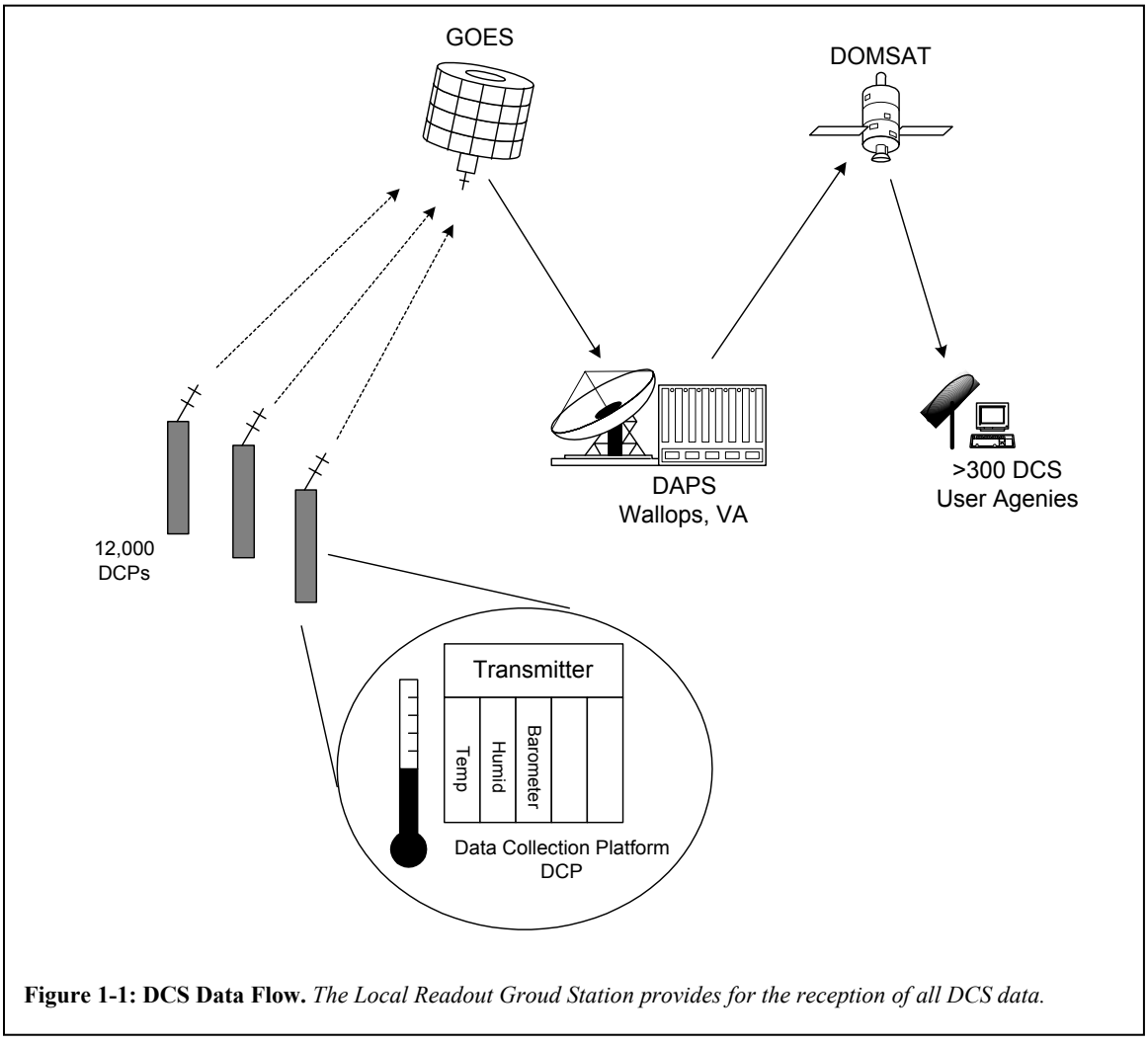


Figure 1-1: DCS Data Flow. *The Local Readout Groud Station provides for the reception of all DCS data.*

messages over DOMSAT using X.25 broadcast-mode protocol at 56 kilobits per second. The Local Readout Ground Station (LRGS) provides user sites with the capability of receiving, archiving, and processing the DOMSAT signal.

The LRGS is designed for continuous operation. It can selectively archive messages from all or any subset of the 100,000 platforms supported by DAPS. The amount of disk space used by the LRGS for DCP messages is limited only by available disk space.

The LRGS supports various means of retrieving the stored data. Users can extract messages via search criteria such as time ranges, DCP addresses, etc. Data can be retrieved via command line interfaces on the LRGS and saved in files. Data can also be retrieved over a network interface.

1.1 The DOMSAT Data Stream

1.1.1 Hardware & Signal Characteristics

The ComStream DBR401 receiver can be configured to generate either RS232 or RS422. Several of the earlier installations have the ComStream generating RS422 and then use a "black box" to convert the signal to RS232. Later installations have the ComStream generating RS232 directly.

RS232 was required by the Franklin Datacom ICP188 board. This is the board being used in all legacy systems to receive DOMSAT data.

Data is received at 56KBPS (synchronous) with an external clock signal provided by the receiver.

1.1.2 DOMSAT Protocols

Three nested protocols are used to transfer DOMSAT messages. DOMSAT is a point-to-multipoint broadcast. No responses are sent from receiver back to the broadcaster.

The broadcast data is encoded using:

- HDLC Lap-B frames
- X.25 Information Packets
- Custom DOMSAT Message Protocol

Figure 1-2 illustrates the protocol nesting.

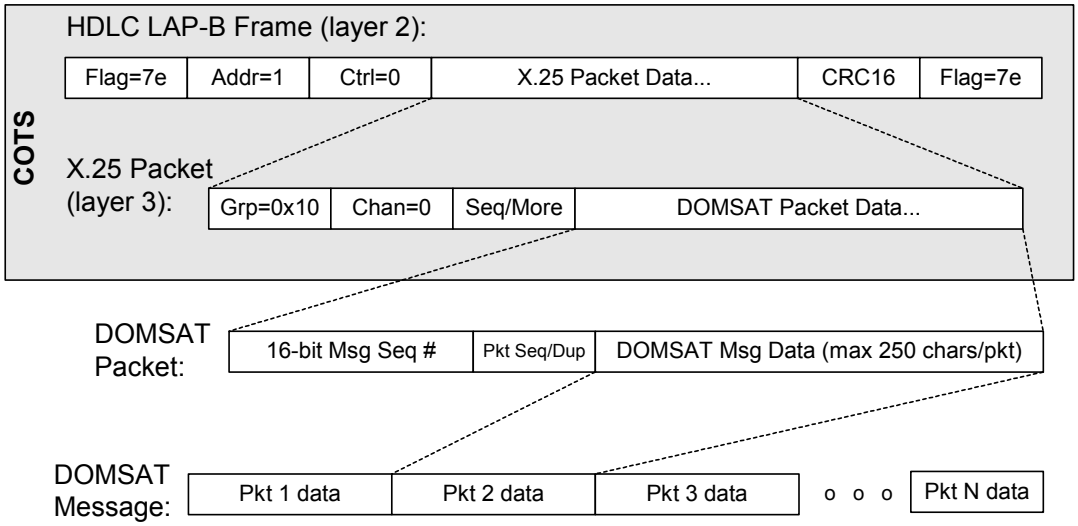


Figure 1-2: DOMSAT Nested Protocols. *Lower levels are handled by the serial board and driver. Upper levels are handled by the LRGS application.*

1.1.2.1 HDLC (LAP-B) Frames

Frames are encoded using a non-standard subset of HDLC LAP-B.

The HDLC address field must always be 1 (or error is assumed and packet is ignored).

The HDLC Control field must always be 0 (or error is assumed and packet is ignored). The high-order bit being 0 indicates that only HDLC information frames are broadcast. Note that the 3-bit sequence number fields and the poll/final bit are always 0. The receiver should not attempt to perform sequence checking or any other processing using these fields.

Zero-bit insertion is used (standard HDLC) and a standard CRC16 is contained in the last two bytes of data.

Each HDLC frame contains an X.25 Packet with a maximum length of 256 bytes.

1.1.2.2 X.25 Packets

Packets are encoded using a non-standard subset of X.25.

The first byte contains various flag bits and a group number. This byte will always have a value of 0x10 (or error is assumed and packet is ignored).

- The high-order bit (bit 7) contains 0, meaning that only data packets are sent (never an interrup packet or a control packet)
- Bit 6 always contains 0. No processing should be done on this bit.
- Bits 5 and 4 always contain 01, indicating that a single byte follows the channel number containing sequence numbers and the 'more bit'.
- The low-order 4 bits always contain the group number of 0.

The X.25 channel field will always be 0. Combined with the 0-group number this means that all data travels on logical circuit 0.

The 3rd byte contains receive/send sequence numbers and the 'more bit'. The sequence numbers are always 0. The receiver should not attempt to perform sequence checking using these values.

The 'more bit' (bit 4) is used to indicate the end of a complete DOMSAT message:

- More=1 means that the message is not yet finished. The receiver should expect more packets to construct the complete aggregate message.
- More=0 means that this is the last packet for a DOMSAT message. The receiver can now process the complete message.

Each X.25 packet contains one DOMSAT packet with a maximum length of 253 bytes.

1.1.2.3 DOMSAT Packet Protocol

Each DOMSAT packet starts with a 2-byte (16-bit) message sequence number. The number is big-endian (i.e. the first byte is the high-order byte). In general, sequence numbers increment with each complete DOMSAT message (which may be composed of multiple packets). Individual packets within a message will have the same sequence number.

Following the message sequence number is a single byte containing:

- A 'Retransmitted Message' Flag (high bit)
- A seven-bit packet sequence number.

The packet sequence number starts at 1 for the first packet within each message, and increments until the complete message is received. There can be a maximum of 64 packets making up a complete DOMSAT message. Hence the largest DOMSAT message can by 64 packets times 250 bytes per packet equals 16000 bytes.

The Retransmitted-Message flag is 0 if this is the first time that this message has been transmitted over DOMSAT. It is 1 if this message has been previously transmitted. Messages are retransmitted in response to a manual user request.

The 16-bit message sequence number starts with 0 (when DAPS is re-booted or a fail-over occurs) and continues to increment. It wraps from 65,535 to 0 and continues to increment.

A DOMSAT packet contains up to 250 bytes of DOMSAT message data.

1.1.2.4 Complete DOMSAT Messages

A complete DOMSAT message is constructed by concatenating the data fields of successive DOMSAT Packets. The maximum message size (imposed by NESDIS, not by the protocol) is 16,000 bytes.

1.1.3 DCP Message Types

There are seven basic types of DCP messages transmitted over DOMSAT:

1. **Normal DCP messages**: These comprise the bulk of the DOMSAT data stream. For normal DCP messages, the DCP address is contained in the message header and the failure code will be 'G' or '?'.
2. **Retransmitted DCP messages**: These are DCP messages which were previously transmitted. They are being transmitted again in response to some user's request or in response to the DOMSAT Quality Monitor (DQM) at Wallops.
3. **DAPS-generated status messages**: In many cases, DAPS will generate a separate status message transmitted immediately after the normal DCP message to indicate some type of failure (e.g. a message received on an unexpected channel). The header will contain the DCP address and the message body will explain the error condition.
4. **Global Bulletins**: The DCP address is hex 11111111. These are from the DAPS administrator addressed to all users of the DCS.
5. **DCP Bulletins**: The DCP address is hex 22222222. These are messages from the DAPS administrator addressed to all users of a particular DCP. The platform address is contained in the first 8 bytes of message data.
6. **Electronic Mail**: The DCP address is hex 33333333. These are from the DAPS administrator addressed to a particular DCS user. The first 8 characters of message data contain the DAPS user-ID of the addressee.
7. **'DAPS-ALIVE' Messages**: The DCP address is hex DADADADA. This message is sent out periodically only if there is no other DOMSAT traffic to guarantee that at least one message will be sent per minute over DOMSAT.

1.1.4 DOMSAT Header Fields

DOMSAT messages begin with a 37 byte ASCII text header followed by a variable length data. The header has the following structure:

- 8 hex digit DCP Address
- YYDDDHHMMSS – Time the message arrived at the Wallops receive station. The day is represented as a three digit day of the year (julian day).
- 1 character failure code
- 2 decimal digit signal strength
- 2 decimal digit frequency offset
- 1 character modulation index
- 1 character data quality indicator
- 3 decimal digit GOES receive channel
- 1 character GOES spacecraft indicator (‘E’ or ‘W’)
- 2 hex digit uplink carrier status
- 5 decimal digit message data length

1.1.4.1 Failure Code

The single character labeled 'Failure Code' in the DCP message header indicates whether the message originated from a DCP or whether it is a DAPS-generated status message. Real DCP messages have a failure code of 'G' for good message, or '?' if the message contained parity errors when received by DAPS.

If the failure code is anything other than '?' or 'G', the message is generated by DAPS. These status messages have the DCP address of the pertinent platform and are typically sent

immediately after the real DCP message from that platform. The body of the message will be a brief text message explaining the event.

Possible failure codes are as follows:

Real DCP Messages:

G Good DCP Message

? DCP Message with Parity Error

DAPS Status Messages:

W Previous DCP message was Received on the wrong channel

D Previous DCP message was duplicated (i.e. received on multiple channels)

A Previous DCP message contained a correctable address error

B Previous DCP message contained a bad (unknown) address

T Previous DCP message was received outside its proper time slice (early/late)

U Previous DCP message was unexpected

M The DCP message for the referenced platform was missing (not received in its proper time slice)

I Previous DCP message had an invalid address

N The referenced platform has a non-complete entry in the DAPS Platform Description Table (PDT)

Q Previous DCP message had bad quality measurements

1.1.4.2 Signal Strength

Signal Strength will be two ASCII digits and will be in the range of 32 to 57. Signal strength is the implied EIRP, assuming the pilot is a +47 dBm reference.

1.1.4.3 Frequency Offset

Frequency Offset will be two ASCII characters. The first will be a plus or minus sign. The second will be on ASCII digit 0 through 9, or the capital letter 'A'. The sign indicates that the DCP is transmitting above or below (plus or minus, respectively) the channel center frequency. The digit indicates the amount of the offset in increments of 50 Hz. The character 'A' represents 500 Hz, which is the worst case frequency error that DAPS can acquire.

1.1.4.4 Modulation Index

Modulation Index will be one of the following three characters:

N Normal: (60° ± 5°)

L Low: (≤ 50°)

H High: (≥ 70°)

1.1.4.5 Data Quality

Data Quality will be one of the following three characters:

N Normal: Error rate better than 10^{-6}

F Fair: Error rate between 10^{-4} and 10^{-6}

P Poor: Error rate worse than 10^{-4}

1.1.5 DOMSAT Data Rates and DAPS Limits

The DAPS was designed around certain assumptions and some absolute limits regarding DCP message traffic. The DRS is designed accordingly to arrive at optimal usage of system resources.

The DAPS can support an absolute maximum of 100,000 platforms. The DRS can support network lists up to this size. Hence any subset of the DOMSAT message stream can be archived. In addition to this, network list filtering can be turned-off completely, to guarantee that the entire stream is recorded.

NESDIS states that DOMSAT through-put is assumed to be nominally 150 megabytes per day. For a simple approximation, we divide 150 megabytes by 100,000 platforms yielding 1,500 bytes per platform per day. An average DRS might be called upon to archive data from approximately 4,000 platforms for a total of 6 megabytes per day. The baseline LRGS devotes 60 megabytes to message storage and can therefore archive 10 day's worth of data.

For a more realistic approximation, DAPS is currently processing only about 15 megabytes per day. Hence even if the entire DOMSAT stream were archived, the baseline DRS can still archive 4 day's worth of data.

1.1.6 The DOMSAT Satellite

The DOMSAT application currently uses the AMC-4 satellite:

- Satellite: AMC-4
- Downlink Frequency: 11830.4 MHz

More information can be found on the web at:

`http://www.ses-americom.com/satellites/amc-4.html`

The DOMSAT EIRP footprint is shown in Figure 1-3

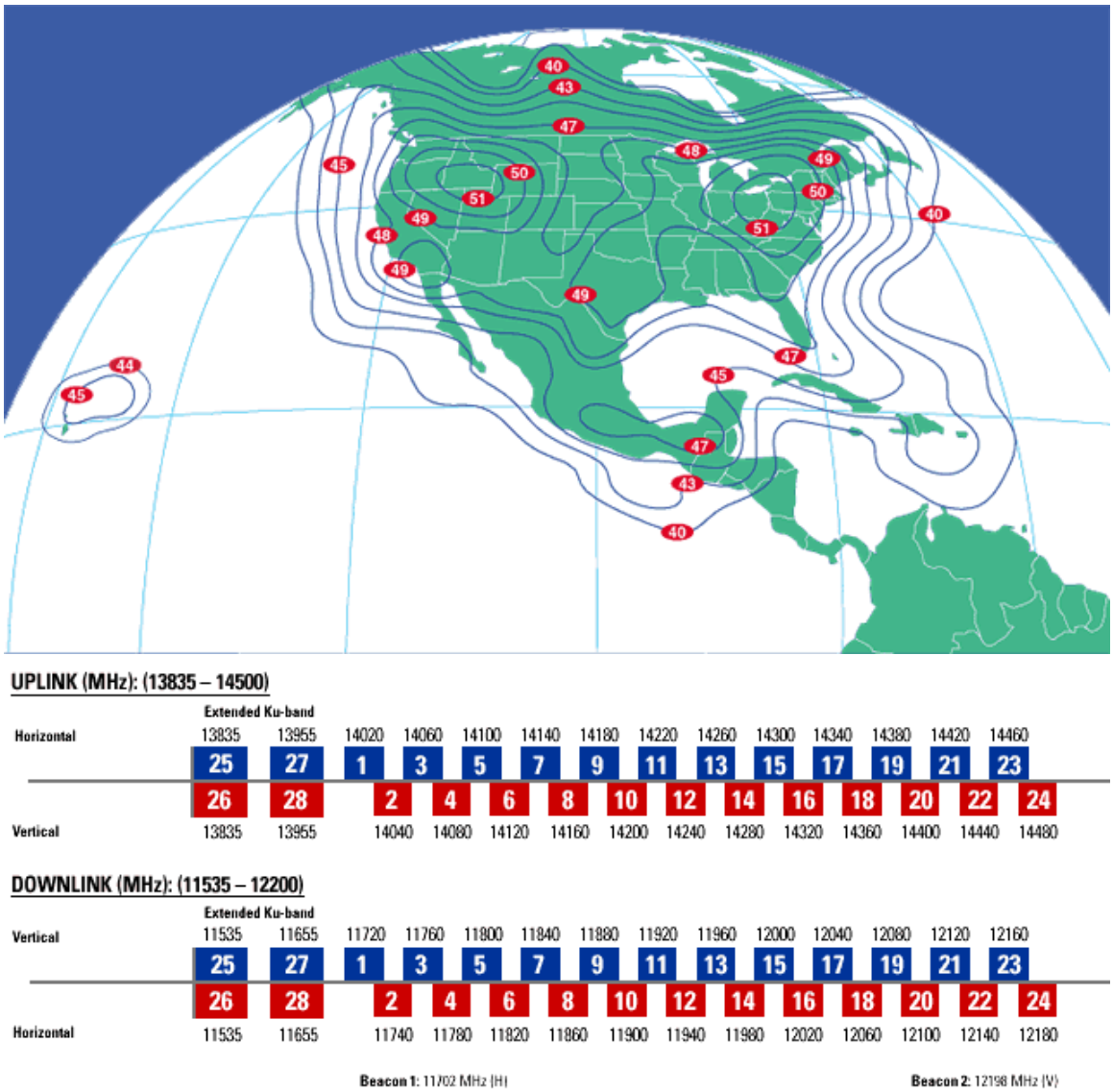


Figure 1-3: DOMSAT (AMC-4) Footprint and Frequency Plan

2. LRGS Theory of Operations

The previous section gives an overview of the entire DRS. This section focuses on the operations of the Local Readout Ground Station.

2.1 Hardware Components

Special hardware components for the LRGS are depicted in Figure 2-1. The Ku-Band satellite antenna collects the Ku-band DOMSAT signal and directs it to the LNB at the antenna focus. The LNB down-converts the signal to L-band and transmits it via coaxial cable to a receiver. The receiver demodulates the signal and outputs synchronous X.25 blocks. The Synchronous Serial Board in the PC decodes the X.25 packets and constructs complete DOMSAT messages.

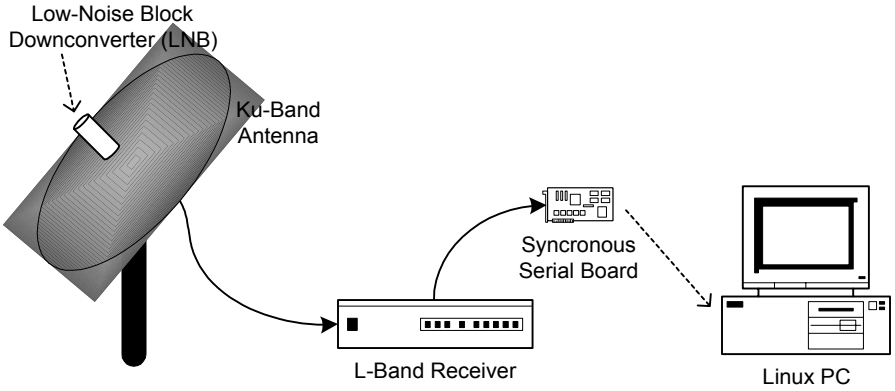


Figure 2-1: LRGS Hardware Components.

The LRGS is hosted on a PC running the Linux Operating System. It is a network ready system with one or more Ethernet interfaces.

The following serial boards are currently supported by the LRGS:

- Franklin Datacom ICP188 (legacy ISA board – no longer available)
- Sangoma S5141
- SDL WANic model 400 or 521 (PCI board for modern systems)
- Emerging Technologies ET5025 (ISA board)

Two different L-Band receivers have been successfully tested with DOMSAT:

- SDM-300L made by ComTech EF-Data (formerly Adaptive Broadband)
- Intellicast 401 made by ComStream (no longer made by the manufacturer).

2.2 Modular Input Design

The LRGS software was designed with expansion in mind. The DOMSAT transmission has now been supported for over 10 years. There are currently other means of acquiring DCP data. These include:

- Direct GOES Receiver
- NOAAPORT Channel 4
- Various internet links

In addition to GOES DCPs, the LRGS might serve as a data collector for other types of devices such as:

- Data collectors that report via terrestrial radio links
- Data collectors that report via cellular or land-line telephone
- Manually collected data

Figure 2-2 depicts how several input interfaces are merged into a single circular data archive.

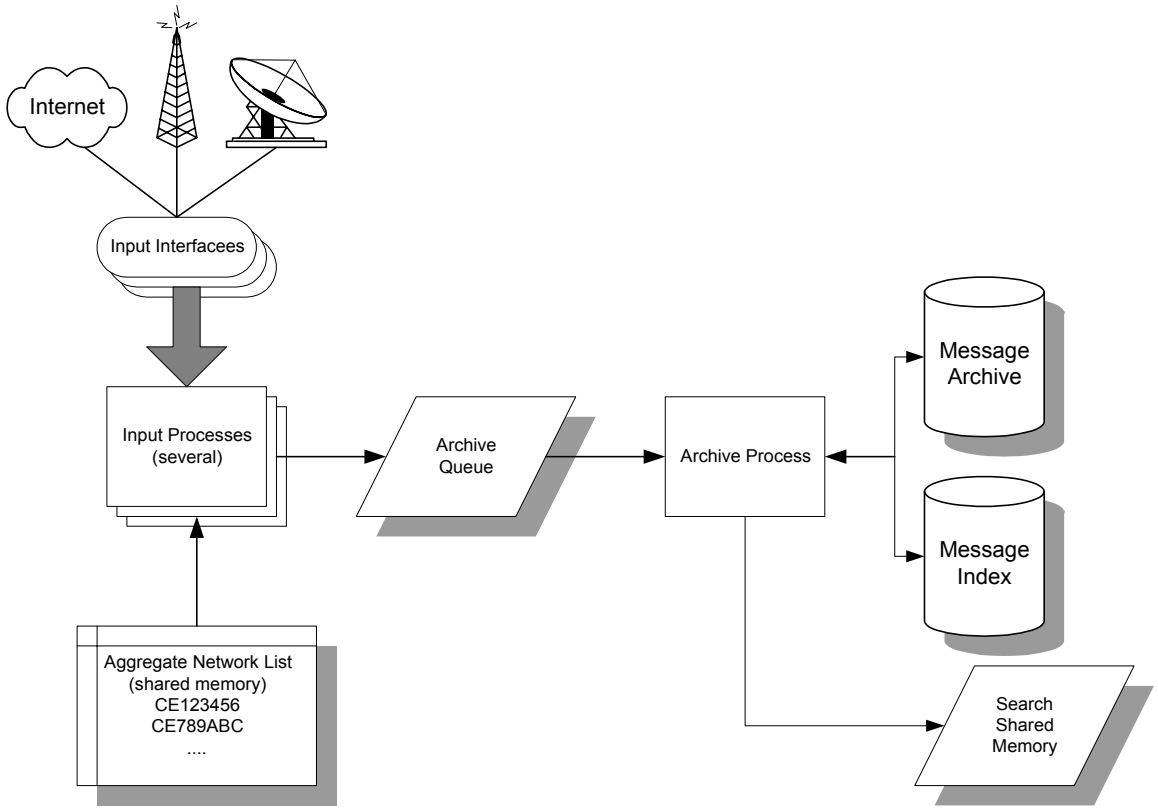


Figure 2-2: Modular Input Interfaces

2.3 Output Interfaces

Messages are stored by the LRGS in a large circular file. The oldest messages are continually overwritten as new messages arrive. An index file of messages is also maintained. The purpose of the index file is to speed time-range and other types of searches.

Users can retrieve data by:

- Extracting data into a file on the LRGS
- Processing data directly on the LRGS as it arrives (via a Java API)
- Retrieving data (either archived or real-time) over a socket-based network link.

2.4 Software Components

The LRGS software is made up of a collection of background programs that work in concert. Each provides a specific function:

- The Control Daemon (lrgs_ctrl_d) initializes the shared system resources. It loads the information found in the "lrgsconfig" file into shared memory. It coordinates the startup and shutdown of other daemons.
- The Archive Daemon (lrgs_archive_d) is responsible for writing DCP messages to the message storage file and maintaining the index. It updates status values in shared memory so that output programs can access the circular files.
- The Event Log Daemon (lrgs_eventlog_d) attaches to the event queue and saves text messages into a log file called "lrgslog". When the log file reaches a predetermined size (10 MB by default), it is renamed to "lrgslog.old" and a new file is opened.
- The Periodic Task Daemon (lrgs_periodic_d) performs various period tasks in the background. It checks each network list that is being used to filter the DOMSAT downlink for changes and reloads new copies when necessary. It also checkes attached processes to make sure they are still running.
- Downlink Daemons (lrgs_sangoma_d, lrgs_wanic_d, lrgs_icp188_d, and lrgs_et_d) receive DCP messages from the various types of serial boards. Most likely you will only have one of these running in your system.
- The ComStream Daemon (lrgs_comstream_d) monitors the ComStream 401 receiver for its status and bit-error-rate while the system is running.
- The DCP Data Server (java lrgs.ldds.LrgsDcpDataSvr) provides a network interface for clients to retrieve messages from this LRGS.
- The Network Backup Server (java lrgs.lnb.LegacyNetbakSvr) allows other systems to use this LRGS as a backup.
- The Network Backup Client (java lrgs.lnb.LegacyNetbakClient) monitors your downlink for gaps and timeouts. When dropouts are detected, it connects to one or more backup LRGS systems and attempts to recover the dropped messages.

In addition to the above, the CORBA network API is made up of the following modules:

- The LRGS Directory Service provides a hierarchical directory of distributed LRGS Systems. Each LRGS uses this to advertise services for this system and (if desired) cooperating LRGS systems.
- The LRGS Advertiser (java lrgs.apicommon.LrgsAdvertise) connects to a list of name services and periodically advertises the services of this system.
- The Event Server (java lrgs.apievents.EventServer) allows clients to monitor the system events being generated on this LRGS.
- The Status Service (java lrgs.apistatus.LrgsStatusServer) allows clients to monitor your LRGS system status.
- The Administrative Service (java lrgs.apiadmin.LrgsAdminServer) allows an authenticated client to perform configuration and control operations on this LRGS over the netwrok.
- The Shared File Digest Server (java lrgs.apicommon.SharedFileDigestServer) allows other LRGS systems to retrieve copies of your network lists that you have declared public.
- The Network List Synchronizer (java lrgs.apicommon.NetlistSyncer) allows you to use network lists from remote LRGS systems.

All of these modules are written in Java and run under a single JVM (Java Virtual Machine).

3. LRGS Installation and Setup

This section explains how to install and configure the LRGS hardware and software components.

3.1 Hardware Setup

The LRGS uses a Linux PC. Such machines are available pre-configured from a number of vendors. This section deals with setting up the non-standard hardware components, the L-Band Receiver and the Synchronous Serial Board.

3.1.1 Installing a Franklin ICP 188 Board

The Franklin ICP188 is used in many legacy DROT and DRS systems, so support for it is provided with the LRGS. These are old ISA boards and are no longer available from the manufacturer.

- Install a Franklin ICP188 board in a spare ISA slot.
- Use the setup screen to reserve the entire 64K of ISA memory starting at D000 for use by the Franklin board.
- Connect custom 'Y' cable to Franklin board on one end (37-pin connector)
- Connect the 'Data' end of the cable to the data port on the ComStream receiver.

The assumption is made that the ComStream receiver is connected to a live DOMSAT downlink.

3.1.2 Installing Sangoma Wanpipe Drivers

The Sangoma S5141 is the preferred synchronous serial board to use with LRGS-DOMSAT.

- Shutdown the system and install Sangoma S5141 board in a spare PCI slot.
- Connect Serial Board to Receiver, both data & remote cables
- Boot Linux
- If a Kudzu screen comes up, tell it to IGNORE the newly-detected device.
- Make a link for kernel development:

```
cd /usr/src
ln –s version linux
```

  … where *version* is the directory containing the kernel.
  for example: ln -s linux-2.4.18-14 linux

- Download latest release, or use wanpipe-2.2.5-3.tgz
- cd /usr/local; tar xvzf *filename* This will create /usr/local/wanpipe.
- Install modules:

```
cd /usr/local/wanpipe
./Setup install
```

- Select the following options:
  - YES: Install Wan Router now.
  - YES Apply kernel patches
  - [enter] to select /usr/src/linux
  - YES: Proceed to build WANPIPE kernel modules
  - YES: Install all modules
  - Yes: Install for current image
  - YES: Install Wanpipe start-up scripts

- Choose default for WANPIPE config files (/etc/wanpipe)
- Likewise for interface files (/etc/wanpipe/interfaces)
- NO: (do not) enable IP forwarding

- Copy the file wanpipe1.conf from the LRGS CD into /etc/wanpipe.
- Reboot the System
- Verify Wanrouter drivers load successfully during boot messages:
    - "Starting wanrouter ….. [OK]"
- The Release CD contains wanpipe.conf. Install this into /etc/wanpipe.

3.1.3 Installing the SDL WANic PCI Board

The WANic 521 is an alternative PCI board.

- Install the WANic board in a spare PCI slot
- Connect the provided DB26 to DB25 adapter cable between the WANic board and the receiver. The 26-pin end connects to the WANic board. The standard 25-pin end connects to the receiver.

3.1.4 Installing an Emerging Technologies ET5025 Board

- Install the ET5025 in a spare slot.
- Use the setup screen to reserve 32K of ISA memory starting at D000 for use by the board.
- Connect a standard straight-through 25-pin cable (commonly called a "switch-box" cable) between the ET5025 and the Data Port on the ComStream receiver

3.1.5 Configure the ComStream Intellicast 401 Receiver

Most existing DOMSAT receive systems use a receiver made by ComStream. The older systems use model DBR401 and the newer systems use an Intellicast 401 receiver. The parameter settings are the similar for both. Refer to the receiver manual for instructions on the serial interface and command syntax. The following settings need to be made:

Parameter	*Setting for DBR401*	*Setting for Intellicast 401*
AL	**1**	**1**
AO	**0**	**0**
B1	**13**	**300**
B2	**2**	**50**
B3	**89**	**2000**
BE	**17**	**0**
BR	**1200**	**1200**
DD	**1**	**1**
DI	**0**	**0**
EE	**1**	**1**
RA	**0**	**0**
RC	**1**	**1**
RM	**0**	**0**
RF	**11830400**	**11830400**
RR	**112000**	**112000**
SD	**1**	**1**

3.1.6 Configure the SDM-300L Receiver

ComStream no longer makes receivers that are compatible with DOMSAT. New systems can use the SDM-300L made by Comtech EF-Data.

Note: The SDM-300L is not as tolerant of signal drift as the older ComStream units were. You must use a PLL (Phase-Lock-Loop) LNB with the SDM-300L. The older DRO (Digital Resonance Oscillator) LNBs that were shipped with DOMSAT systems will not work.

The SDM-300L receiver can be configured via the push-buttons on the front of the chassis. Refer to the manual for details on navigating the menus. The following settings need to be made.

- FUNCTION SELECT UTILITY
  - Utility Fixed Modem Rate
    - CRDR: BPSK 1/2 56 KBPS
  - Utility Interface
    - RX TERR Interface: RS232
  - Utility Demod
    - Rx Terminal LO 0 MHz Mix:+
    - Demodulator Type CSC CLOSED
    - Decoder Type: SEQUENTIAL
    - RX BPSK Ordering: STANDARD
    - Demod Spectrum: INVERT
- FUNCTION SELECT CONFIG
  - CONFIG DEMOD
    - RX-A BPSK 1/2 56 KBPS
    - RX-IF Freq: 1080.4000 (Ku band downlink is 11,830.4 MHz, converted to 1080.4 MHz by the LNB)
    - Descrambler: On
    - Differential Decoding: On
    - IF Loopback: Off
    - BER Threshold: None
    - Sweep Center: 0
    - Sweep Range: 150000
    - Reacquisition: 0
    - LNB Power: ON
    - LNB Voltage: 18
    - LNB 10MHz REF: Off
  - CONFIG INTERFACE
    - Ext Clock Freq: 1544 Khz
    - Buffer Clock RX: (Satellite)
    - RX Clock Phase: Normal
    - B-Band Loop Back: Off
    - Buffer Size: 1024
    - Loop Timing: Off

3.1.7 Connect the Receiver for M&C Functions

Use a standard RS232 serial cable to connect COM1 on the PC to the Monitor & Control (M&C) port on the receiver. On ComStream receivers this port is labeled “M&C”. On SDM-300L receivers it is labeled “Remote”.

3.2 Linux Installation

Computers are available from a variety of vendors with Linux pre-installed. The base installation that these vendors provide is usually sufficient to support the LRGS.

If you need to install Linux yourself, do a "Custom" installation of Linux on the system. In addition to the pre-selected package groups for a normal Gnome-Terminal, select the following groups:

- DOS/Windows Connectivity
- Development
- Kernel Development
- NFS Server
- Documentation
- Web Server
- emacs Text editor
- Tools

3.3 Install Network Time Protocol

Login as root on the Linux console. Place the RedHat Linux Installation Disk in the CD Drive. Then type:

```
mount /dev/cdrom
```

Then open a file-browser. Navigate to /mnt/cdrom/RedHat/RPMS and double click on the file named "xntp3-5.93-13.i386.rpm. This will install the Network Time Protocol daemon.

Next, edit the file /etc/ntp.conf. Add server lines for the NTP servers that you wish to use. For example:

```
server clock.psu.edu
server ntp-2.vt.edu
server clock1.unc.edu
```

You can find a list of public NTP servers at

http://www.eecis.udel.edu/~mills/ntp/servers.htm.

Start the NTP daemon on your system by typing:

```
chkconfig –level 2345 ntp on
```

Finally, unmount the CD ROM by typing

```
umount /mnt/cdrom
```

3.4 Mount the Release CD-ROM

The Distribution CD-ROM contains the LRGS Release, the Java Runtime Environment, The JacORB Object Request Broker and other files needed for installation.

On many machines, the CD drive is mounted automatically when you place a disk into it. If you need to mount the CD manually, place the LRGS distribution CD into the drive. Login as root. Then type:

```
mount /dev/cdrom
```

This will mount the CD with default arguments and at the default mount point of /mnt/cdrom. You may see a warning message about the device being write-protected. This is OK.

Verify that you can read the contents of the CD by typing:

```
ls /mnt/cdrom
```

You should see a directory listing of files.

3.5 Install and Configure the Java Run-Time Environment (JRE)

The Java Runtime Environment must be installed in order for the LRGS software to run. Go to

```
http://java.sun.com/
```

Use the links to navigate to the latest SDK (Software Development Kit) for Linux RPM. At the time of this writing, this will be a file called:

```
j2sdk-1_4_1_03-linux-i586-rpm.bin
```

Install the SDK as per the instructions from Sun. This will result in a directory called /usr/java. The LRGS expects a link called /usr/java/jdk. Construct this link as follows:

```
cd /usr/java
ln -s release-dir-name jdk
```

3.6 Create a Login Account for 'lrgs'

The 'lrgs' login account is used for administration of the software. All of the background processes run as user 'lrgs'.

After Linux is installed, use the 'linuxconf' program to create a new user account.

- Login Name: lrgs
- Home Directory:
  - Some system administrators prefer to place user accounts under /usr and some prefer /home. Determine how Linux was configured on your system and select a home directory (probably either /home/lrgs or /usr/lrgs) that has a generous amount of disk storage.
- Supplemental Groups: Type in 'root'.
- All other fields: Use default values.

3.7 Install the LRGS Distribution

The LRGS installation is contained in a the file with the name:

```
lrgs-version.i386.rpm
```

…where *version* is the release number. For example, the 3.4-0 release is contained in the file:

```
lrgs-3.4-0.i386.rpm
```

With the CD mounted, as described above, login as root and type:

```
rpm –i <filename-as-above>
```

(Note the two hyphens before the word "force". It must be typed exactly as shown above.)

Installing the LRGS will not overwrite configuration files from a previous release, if any are found.

3.8 Setup Administrative Users

In order to be able to run administrative tasks over the network, you will have to setup a password file on this LRGS. Login as LRGS and create the file the first time by typing:

```
cd ~
touch .lrgs.passwd
chmod 600 .lrgs.passwd
```

Now run the utility for adding, removing, and modifying administrative users. A simple command-line utility is prepared for this. Start the utility by typing:

```
editPasswd
```

Type 'help' and hit ENTER for a list of supported commands. For example, to add an administrative user named 'emily', type:

```
adduser emily
...(type the password as you are prompted)
addrole emily admin
write
quit
```

3.9 Setup the System for your Downlink Card

The LRGS supports several different downlink cards for receiving the X.25 DOMSAT data stream. You must load the appropriate driver for the board installed in your system. You must login as root in order to have permission to install drivers.

After installing drivers, remember to modify your "lrgsconfig" file to start the correct data collection daemon.

If your system has a Sangoma S5141 card, then the driver is loaded automatically when the system boots. You do not need to do this explicitely.

If your system has an SDL WANic card, login as root and type:

```
cd ~lrgs/bin
./load_WANic
./boot_WANic
```

If your system has a Franklin Datacom ICP188 card, login as root and type:

```
cd ~lrgs/bin
./load_icp188
./boot_icp188
```

If you system has an Emerging Technologies ET5025 card, login as root and type:

```
cd ~lrgs/bin
./load_et5025
./boot_et5025
```

In all cases, the 'load_…' command loads the driver immediately. The 'boot_…' command sets up the system so that the driver will be loaded automatically whenever the system is booted.

In some cases, the load_ command will examine the version of the Linux kernel that you are running and choose an appropriate driver.

4. LRGS Run-Time Environment

4.1 The LRGS Administrative Account

Prior to installing the software, you created a UNIX user account called 'lrgs'. This account is the owner of all of the runtime files and processes. You use this account for administrative purposes.

4.2 The 'lrgsenv' File

A shell-script called 'lrgsenv' is placed in the 'lrgs' user's home directory when the software is installed. This script contains several settings that control the behavior of the LRGS.

4.2.1 General Environment Settings

The following environment variables MUST be set:

- JDKDIR – This is the location of the installed Java virtual machine. In most cases, it should be set to /usr/local/jdk
- ORBDIR – This is the location of the Object Request Broker that was installed with the LRGS. In most cases it should be set to /usr/local/JacORB
- PATH – This setting adds the Java and ORB to your command interpreter path.
- CLASSPATH – This setting tells the object request broker where to look for Java components. It is a list of modules, separated by colons. It must have the following files in it:
  - /usr/local/bin/lrgs.jar – Java components for the LRGS.
  - /usr/local/bin/ilex.jar – Java components for the generic Ilex library.
  - $ORBDIR/lib/jacorb.jar – Java components for the object request broker.
  - /usr/local/bin/xercesImpl.jar - XERCES XML Parsing classes.
  - /usr/local/bin/xercesSamples.jar - ditto
  - /usr/local/bin/xmlParserAPIs.jar - ditto

4.2.2 Settings that control the LRGS API

API Settings are in two places:

- Environment Variables set on login in the 'lrgsenv' file.
- Properties set in ~lrgs/lrgsapi.properties

Previous version of the API used several variables that were set in the lrgsenv file. In LRGS 3.2, there is only one:

```
API_ARGS
```

This contains arguments to pass to the API executable when it is started. The only useful value would be to set the debugging level via –d1 (least verbose, -d2, or –d3 (most verbose).

The API properties in the file lrgsapi.properties are listed in section 6.4.

4.3 Message Files

The LRGS Configuration File specifies the name of the files in which message data is stored. By default, the following names will be used:

- ~lrgs/messages – Circular storage of actual message data.
- ~lrgs/messages.I – Index file used for fast searching.

4.4 The Log File

The LRGS background processes generate events to a shared memory queue. To view a scrolling list of events, type:

```
showevents
```

The LRGS event log daemon process constantly reads events and writes them to the file ~lrgs/lrgslog. The file grows to a pre-determined size. When the file becomes full it is closed and renamed to ~lrgs/lrgslog.old. Then a new file is opened. By default, the predetermined size is 10 MB.

There are several priorities for events. Each event is tagged with one of the following:

- DEBUG3 – low-level voluminous debug info
- DEBUG2 – mid-level fairly voluminous debug info
- DEBUG1 – high-level not-so-voluminous debug info
- E_INFORMATION – normal processing
- E_WARNING – abnormal but recoverable condition
- E_FAILURE – a requested operation could not be performed
- E_FATAL – a fatal error occurred in an LRGS process (the process subsequently aborted).

All LRGS processes can be started with a debug flag (-d1, -d2, or –d3). The flag determines the amount of debug-information that will be recorded as events.

4.5 IOR Files

CORBA objects are uniquely identified by an Interoperable Object References (IOR). These IORs can be converted into printable strings and stored in ASCII files. These files can then be used as arguments to other programs that need to connect to these objects.

The following IOR files are created by the LRGS software. All files are stored in the LRGS home directory.

- LrgsDirectory.ior — Directory object for advertising this and other LRGS Systems.
- LrgsSystem.ior — Object in which this LRGS system advertises its services
- api_internal.ior — Object used for local inter-process communication

4.6 DCP Data User Directories

Each DCP Data User will have a sub-directory for local storage of network list and search criteria files. The Configuration File specifies the root directory where these are stored.

The default location is ~lrgs/users. Hence if an LRGS has three accounts for moe, larry, and curly, the following directories will exist:

- ~lrgs/users/moe
- ~lrgs/users/larry
- ~lrgs/users/curly

Note that DCP Data Users do not require Unix user accounts.

5. The LRGS Text-Mode User Interface

All LRGS functions can be performed in text-mode (as opposed to the GUI programs described later). The text mode utilities involve opening a Unix shell window.

Unlike the Network API functions described later, these utilities operate directly on the local LRGS system.

5.1 Starting All LRGS Services

A script called 'lrgsAutoStart' is included with the LRGS distribution. This script is executed by the Administrative GUI program when you press the "Normal Startup" button. You can also execute this script from the command line by typing:

```
lrgsAutoStart -f
```

The auto-start script reads a file of parameters that you can edit. The file is called 'lrgsAutoStartParams'. A typical files looks as follows:

```
#Fri Sep 08 16:49:38 EDT 2000
DoNormalStart=Y
StartLNBS=Y
StartLDDS=Y
EventLogSize=2000000
EventLogFile=~lrgs/lrgslog
StartEventLogger=Y
StartDomsat=Y
StartLNBC=Y
```

Most fields are either 'Y' or 'N':

- StartLNBS – Y/N: Start the Legacy Network Backup Server
- StartLNBC – Y/N: Start the Legacy Network Backup Client
- StartDomsat – Y/N: Enable the DOMSAT link after starting all daemons.
- StartEventLogger –Y/N: Start the event-log daemon to log runtime events to a file.

The 'lrgsAutoStartParams' file is maintained by the LRGS Administrative GUI and should probably not be edited directly, unless you do not plan to use the GUI.

5.2 Administrative Functions

Administrative functions include starting and stopping the LRGS software components, and modifying the LRGS configuration.

5.2.1 The 'lrgs' Command Line

A simple command-line utility called 'lrgs' can perform most software control functions. Type 'lrgs' followed by the options listed below:

```
lrgs start
```

This command starts the LRGS control daemon, which in turn initializes the LRGS shared resources and starts the core daemons. Core daemons are the processes listed in the 'lrgsconfig' file with the KID keyword.

```
lrgs stop
```

This command stops the LRGS control process and the core daemons.

```
lrgs domsat enable
```

```
lrgs domsat disable
```

These commands tell the DOMSAT interface daemon to start or stop receiving DCP messages. It is sometimes useful to disable and then re-enable the link in trouble-shooting situations.

```
lrgs datasvr enable
```

```
lrgs datasvr disable
```

These commands start or stop the LRGS DCP Data Server. Network clients will not be able to pull messages from your system if this server is disabled.

```
lrgs netback c enable
```

```
lrgs netback c disable
```

These commands start or stop the network backup client. This process checks for dropouts or timeouts in the DOMSAT link and attempts to recover data from cooperating LRGS systems.

```
lrgs netback s enable
```

```
lrgs netback s disable
```

These commands start or stop the network backup server. This process allows other LRGS systems to use your system as a backup.

```
lrgs netlist
```

This command will cause all network lists that are being used to filter the downlink to be reloaded. It is usually not necessary to do this because the LRGS Periodic Daemon will check for changes once per minute and reload automatically.

5.2.2 Modifying the LRGS Configuration

The LRGS configuration is kept in a text file in the LRGS home directory. The file is called "lrgsconfig". You can edit this file with any text-editor. The file format is described in section **Error! Reference source not found.**.

If you edit the file while the LRGS is running, changes will not take effect immediately. You must load the changes by typing:

```
reload_config
```

You do not need to do this if the LRGS is *not* currently running. The changes will be loaded when you next start the software.

5.3 Viewing DCP Messages

The 'showmsg' program reads messages directly out of the the LRGS storage files and prints them to the standard output. Start the program as follows:

```
showmsg options ...
```

The following options are recognized:

-f *filename*	Evaluate the contents of the named search-criteria file and print only the messages that meet the criteria.
-n *name*	When connecting to the LRGS resources, use the specified name. The 'showprocs' utility (see section 5.5) will show this name as the attached process.
-N *number*	Only print this many messages. Then quit.
-b *prefix*	Print this string before each message. The string can contain standard UNIX escape sequences (such as \n, \r, \f, \001).
-a *suffix*	Print this string after each message. The string can contain standard UNIX escape sequences.
-v	Print a verbose header for each message containing the message sequence number and LRGS receipt-time.

5.4 Viewing System Events

The 'showevents' program connects to the shared-memory queue of LRGS runtime events and prints them to standard output. Start the program as follows:

```
showevents
```

or

```
showevent backlog
```

… where *backlog* is the number of past events to display. The default value is 20.

The showevents utility will continue to run, printing events as they are generated. You can stop it by pressing CTRL-C.

5.5 Viewing Attached Processes

The 'showprocs' utility connects to the LRGS shared memory and prints a list of client processes that are currently attached. Start the program as follows:

```
showprocs [-u update-period] [-c]
```

The –u and –c arguments are optional. If –c is given, showprocs will clear the screen before printing the list. If –u is given, showprocs will run continuously, printing a new list at the specified period (in seconds). The default is to print the list once and exit.

A typical use is to continually print the list every second, clearing the screen each time. This gives the effect of a dynamically updating display:

```
showprocs –u1 –c
```

5.6 Trouble Shooting Utilities

These utilities will not be needed during normal operations, but may be useful for trouble-shooting.

5.6.1 Cleaning IPC Resources

The LRGS uses several UNIX shared memory and semaphore resources to communicate between the various background processes. The ‘cleanipcs’ utility can be run to remove these resources.

Do not run cleanipcs when the LRGS is running.

```
cleanipcs [-a]
```

If –a (all) is given, cleanipcs removes all resources. Even the ones normally left in place between LRGS runs.

5.6.2 Franklin ICP188 Diagnostics Program

If you suspect problems with your Franklin ICP188 board (assuming you have one), you can use the icp188diag utility to perform several low-level diagnostics. Start the utility as follows:

```
icp188diag [device-name]
```

The device-name argument is optional. If omitted, it will use the default device name of /dev/icp188.

Once started, press ENTER for a list of supported commands.

6. Network Application Program Interface

The LRGS provides a network-accessible application program interface (API) so that you can perform status monitoring, configuration, and control operations from a remote site. The LRGS is typically configured to start the API when it boots.

A 100% pure Java GUI is provided that uses the network API. The GUI is packaged separately to make it easy to download to any machine.

Using the GUI and the API you can monitor LRGS status and perform maintenance functions from any network computer, for example, a Windows NT machine on your desktop.

6.1 Theory of Operations

The Network API is made up of several components. Each component runs as a thread within a single JVM (Java Virtual Machine). Each component provides services via a CORBA interface. The components are:

- **LRGS Directory Service –** Provides a directory of cooperating LRGS systems. Your LRGS and cooperating LRGS systems will 'advertise' themselves in this directory.
- **LRGS System –** Top-level service for a particular LRGS system, providing access to all of the monitor and control objects (listed below).
- **LRGS Event Service** – Provides external clients with the ability to retrieve a time-tagged list of events that are generated by this LRGS. These events are very useful in trouble-shooting situations.
- **LRGS Status Service** – Provides external clients with the ability to monitor several status variables having to do with archive statistics, downlink status, etc.
- **LRGS Administrative Service** – Requires proper authentication (user name and password). This service allows you to remotely configure and control the LRGS.
- Shared File Digest Server – Allows you to share your network lists with the network community. Other LRGS users will be able to download and use copies of your network lists.
- **Network List Synchronizer** – Allows this LRGS to use remote network lists that are maintained at other sites. The program checks for changes periodically and downloads a new copy when necessary.

6.2 Starting and Stopping the API

Three shell scripts have been prepared to make it easy to start and stop the API:

`startAPI [-i]`
: This starts the CORBA API.

`showAPI`
: Uses the UNIX 'ps' command to show the process IDs of the running API. Under Linux, each thread shows up as a separate process ID. There are typically 15 to 20 threads running under the API.

`killAPI [-i]`
: Kills all components of the CORBA API.

6.2.1 Java Command Line to start API

As a reference, here is the Java command line used within the 'startAPI' script:

```
java lrgs.directory.APIServer [options]
```

The API gets most of its parameters from a properties file (see below). The only meaningful command line argument is to set the debug level (off by default):

```
-d1
-d2
-d3
```

6.3 LRGS Directory Service Replaces CORBA Name Service

Previous versions of the LRGS used the CORBA Name Service to create a distributed community of cooperating LRGS sites. Using the CORBA Name Service to build a directory of distributed LRGS systems had several drawbacks:

- Some implementations of the standard service are slow and unstable.
- Collecting status from each system in the directory involved making CORBA connections to each system. Consequently the GUI menu of systems took a long to to come up.

A new LRGS Directory service has been added to solve these problems.

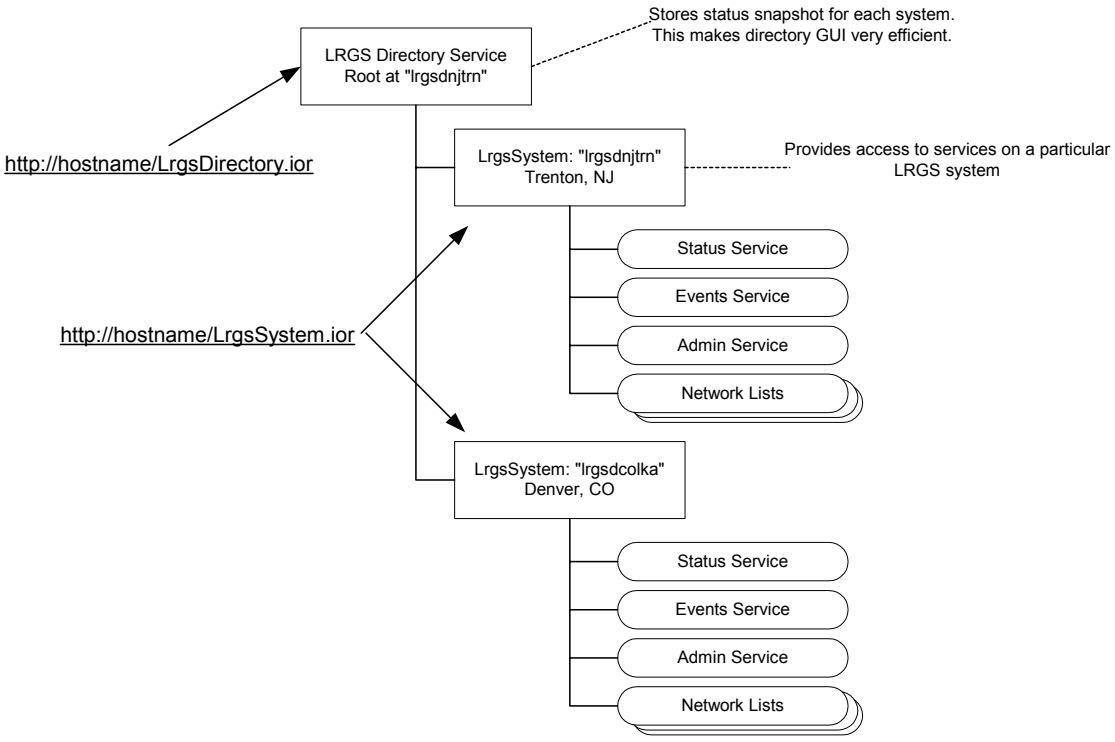


Figure 6-1: LRGS Directory Layout

As depicted, each LRGS provides a directory service, in which LRGS systems can advertise themselves. In each directory will be the local LRGS and also any remote LRGS systems that are cooperating with you.

6.3.1 IOR Files

The LRGS API creates two files. These files hold string versions of the CORBA interoperable object references (IOR):

LrgsSystem.ior: Reference to an object containing all of the services running on this LRGS. By accessing this object, a process can gain access to all of these services.

LrgsDirectory.ior: A directory containing references to LrgsSystem objects. Cooperating systems will advertise their services in each other's directories.

These two files are created in the LRGS home directory and are owned by LRGS. When the LRGS is installed, symbolic links are made from the HTTP (web) directory to these two files. Hence these files are available over the web, enabling external processes to connect.

An additional IOR file called **api_internal.ior** is created in the LRGS home directory. This is a reference to an object for internal use only. It should not be exported to the web.

6.4 API Properties

When the API starts it reads a properties file in the LRGS home directory. The name of the file is "lrgsapi.properties". This file replaces most of the settings that were formerly stored in the shell script 'lrgsenv'.

Property Name	Default Value	Description
hostname	Defaults to fully-qualified domain name.	The name used to advertise this system in other directories.
dir00 … dir99	(none)	Up to 100 URLs, each pointing to an external LRGS directory service accessible over the network. This LRGS will advertise its services in each of these directories. You do not need to include the URL to this local LRGS, services will be advertised there automatically.
advertisePeriod	120	Number of seconds between attempts to advertise this LRGS's services in external services.
netlistSyncPeriod	600	Number of seconds between attempts to synchronize network lists.
sweepAge	300	Every external system must update its status in the directory within this period. Otherwise the reference to it is removed.

You can edit the properties file when the API is up. You then can activate your changes by typing the following command:

```
java lrgs.directory.resetapi
```

Look at the events for confirmation that the reset was successful:

```
showevents
```

6.5 The Administrative Password File

In order to grant administrative privileges to remote clients, a password file called ".lrgs.passwd" must exist in the LRGS home directory.

This file should be kept protected! Its permissions should be read/write for user LRGS (the owner) but nothing for everyone else.

Create the file by logging in as LRGS and typing the following:

```
touch .lrgs.passwd
chmod 600 .lrgs.passwd
```

A command-line Java program has been prepared for you to edit this file. Run the program by typing:

```
editPasswd
```

To see a complete list of available commands in the editor, type ‘help’. To add an administrative user named “joe” with a password “aj3-fb5”, issue the following commands:

```
adduser joe
(you are then prompted for the password)
addrole joe admin
```

Verify that Joe now exists and has administrative privileges by typing:

```
show joe
```

Finally, write the file and quit:

```
write
quit
```

6.6 The 'lrgsenv' File

Most of the API settings that were formerly placed in the lrgsenv file have been moved to the API properties file. The following settings are still supported:

API_ARGS

Normally this is set to an empty string. For troubleshooting you can set it to –d1, -d2, or –d3 for various levels of verbosity in debugging events.

6.6.1 How to Disallow Other Systems from Advertising on Your Machine

When the API Directory Service is started from the "startAPI" script, it creates an IOR file to the directory object. This file is called "LrgsDirectory.ior" and is placed in the LRGS home directory. To check for this file, type:

```
ls –l ~lrgs/LrgsDirectory.ior
```

When the LRGS is installed, it places a symbolic link to this file in the public HTML directory. It does this only if you have the web server installed on your system. To prevent outside users from advertising on your LRGS, simply remove this link. First login as root and then type:

For RH Linux 6.2

```
rm /home/httpd/html/LrgsDirectory.ior
```

For RH Linux 7.0 and later

```
rm /var/www/html/LrgsDirectory.ior
```

6.6.2 How to Advertise this LRGS Service on other Systems

The properties file contains entries of the form:

```
dir00=URL
dir01=URL
…
```

These specify the external systems where your LRGS will be advertised.

Each entry must have a unique name, this is the purpose of the two-digit extension.

After the equals sign, place the URL pointing to the IOR file for the directory on the remote system. For example, to advertise at the wallops machine cdadata.wcda.noaa.gov:

```
dir01=http://cdadata.wcda.noaa.gov/LrgsDirectory.ior
```

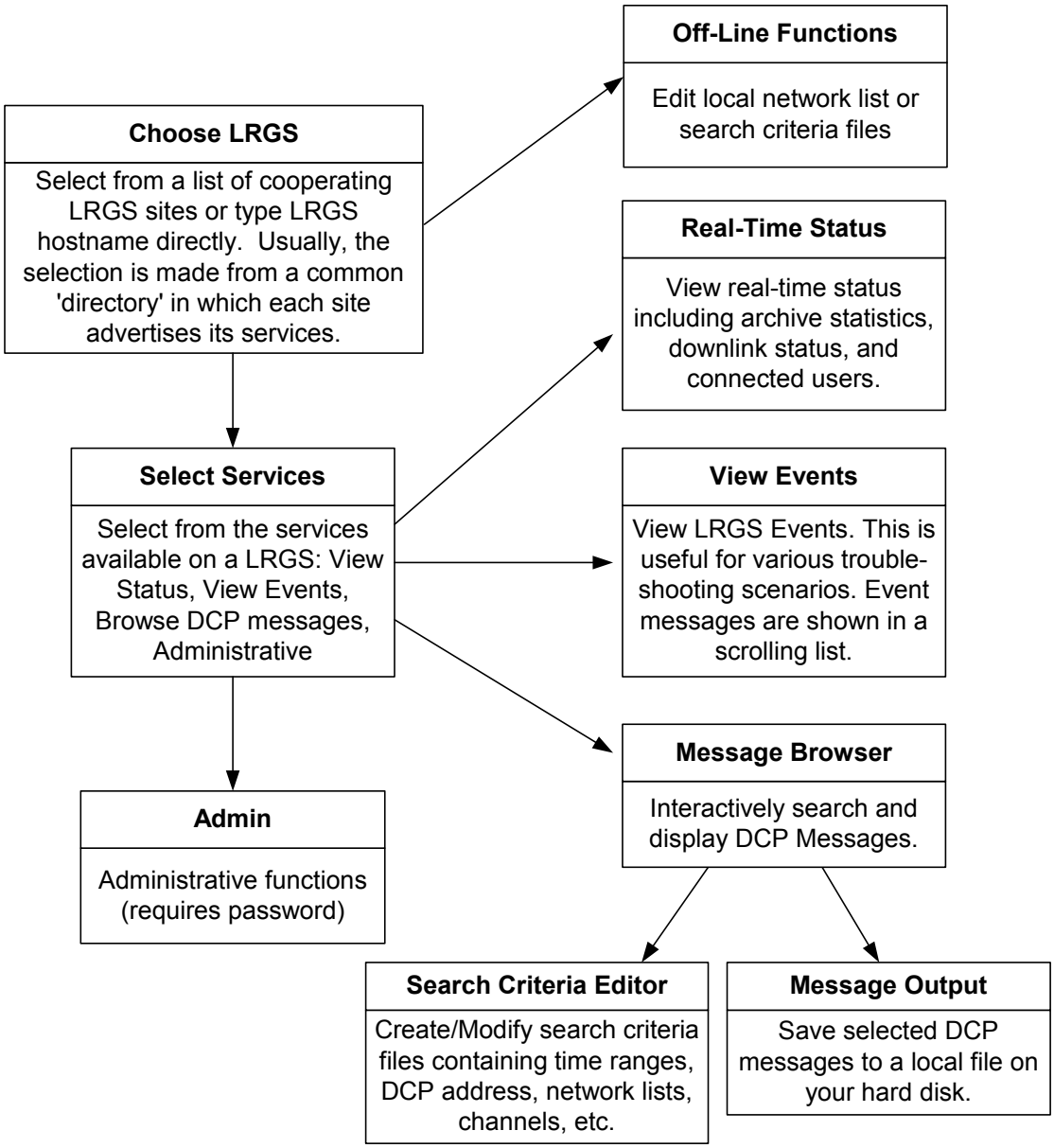
7. Graphical User Interface (GUI)

Figure 7-1: LRGS GUI Hierarchy.

The LRGS client GUI is a hierarchical series of screens. The flow from one screen to another is depicted in Figure 7-1. Each box in the figure represents a screen in the GUI.

You can enter the GUI at any point by starting the appropriate Java class with required command-line arguments. Shell scripts (or "Batch Files" if you're running Windows) are provided for starting the "Choose LRGS", "Select Services", and "Message Browser" screens.

You can find reference manual pages for the GUI screens at the Ilex Engineering web site: http://www.ilexengineering.com/LRGS-3.2/help/contents.html.



7.1 Installing the Client Components

The LRGS installation described above installs all client components on your Red Hat Linux machine. This section describes how to run the portable GUI on other machines.

The client components are 100% pure Java and can run on any machine with a Java Runtime Environment (JRE) supporting Java Version 1.2 or later. The client also requires archives of the JacORB Object Request Broker.

The client code has gone through several iterations. As of this writing, the client code is bundled with DECODES. When you download the client, make sure you also download the accompanying LRGS Client Guide document. This will contain instructions appropriate to the version you are using.

7.2 Configure the GUI

Start up scripts are included in the LRGS Client package to start up the GUI. By default, the scripts will connect to the LRGS systems whose services are advertised on the Wallops machine cdadata.wcda.noaa.gov. The scripts also contain environment variables specifying the locations of the Java Runtime Environment (JVM) and Object Request Broker (ORB).

The scripts will work fine in most cases as downloaded. Some reasons to modify the scripts might be:

- You are running the Generic LRGS Client on a non-Linux Unix machine and need to change settings for the JVM and ORB.
- You are hosting your own group of cooperating LRGS systems that are not advertised at Wallops.

For UNIX Systems that already have a JRE installed, modify the PATH and JRE_JAR variables defined in "javaenv.sh" appropriately.

7.3 Configure the GUI Properties

The LRGS GUI reads a properties file from the LRGS installation directory to control various actions. When the GUI starts, it looks for the properties file called "lrgsgui.properties" in several directories in the following order:

- Specified directory (if a full path name was given with the –P option)
- Current working directory
- The user's home directory
- All of the directories listed in the CLASSPATH environment variable

If no properties file is found in any of these directories, a warning message will be issued and reasonable default values will be chosen.

The following table lists the supported properties.

Property Name	*Value Type*	*Description*
General.DirectoryIOR	URL	Points to the file containing the interoperable object reference (IOR) to the LRGS directory service.
General.HelpContents	URL	HTML file that is displayed on any screen when user selects 'Contents' from the help menu.
General.HelpAbout	URL	HTML file that is displayed on any screen when user selects 'About' from the help menu.
General.Browser	command	The command used to spawn a web browser. This is used for the help system.
General.TextEditor	command	The command used to spawn a text editor. If you leave this property blank the system will attempt to select an editor appropriate to your platform.
LrgsAccess.height	Integer	Preferred height (in pixels) for the LrgsAccess panel running on this machine. Set blank to revert to default.
LrgsAccess.width	Integer	Preferred width (in pixels) for the LrgsAccess panel running on this machine. Set blank to revert to default.
LrgsAccess.x	Integer	Preferred horizontal position (X coordinate) for the LrgsAccess panel running on this machine. Set blank to revert to default.
LrgsAccess.y	Integer	Preferred vertical position (Y coordinate) for the LrgsAccess panel running on this machine. Set blank to revert to default.
LrgsServices.x	Integer	Preferred horizontal position (X coordinate) for the LrgsServices panel running on this machine. Set blank to revert to default.
LrgsServices.y	Integer	Preferred vertical position (Y coordinate) for the LrgsServices panel running on this machine. Set blank to revert to default.
LrgsServices.height	Integer	Preferred height (in pixels) for the LrgsServices panel running on this machine. Set blank to revert to default.
LrgsServices.width	Integer	Preferred width (in pixels) for the LrgsServices panel running on this machine. Set blank to revert to default.

Property Name	*Value Type*	*Description*
LrgsServices.Help	URL	HTML file that is displayed when user selects 'This Screen' from the help menu.
LrgsServices.AdminEnabled	true/false	If set to 'false', the "System Administration" button is not displayed on the GUI. The default is 'true'.
RealTimeStatus.x	Integer	Preferred horizontal position (X coordinate) for the RealTimeStatus panel running on this machine. Set blank to revert to default.
RealTimeStatus.y	Integer	Preferred vertical position (Y coordinate) for the RealTimeStatus panel running on this machine. Set blank to revert to default.
RealTimeStatus.height	Integer	Preferred height (in pixels) for the RealTimeStatus panel running on this machine. Set blank to revert to default.
RealTimeStatus.width	Integer	Preferred width (in pixels) for the RealTimeStatus panel running on this machine. Set blank to revert to default.
RealTimeStatus.Help	URL	HTML file that is displayed when user selects 'This Screen' from the help menu.
RealTimeStatus.UpdatePeriod	Integer	Number of seconds between automatic updates to the real-time status display. Set to 0 to disable automatic updates.
LrgsControl.Help	URL	HTML file that is displayed when user selects 'This Screen' from the help menu.
SearchCritEditor.height	Integer	Preferred height (in pixels) for the SearchCritEditor panel running on this machine. Set blank to revert to default.
SearchCritEditor.width	Integer	Preferred width (in pixels) for the SearchCritEditor panel running on this machine. Set blank to revert to default.
SearchCritEditor.x	Integer	Preferred horizontal position (X coordinate) for the SearchCritEditor panel running on this machine. Set blank to revert to default.
SearchCritEditor.y	Integer	Preferred vertical position (Y coordinate) for the SearchCritEditor panel running on this machine. Set blank to revert to default.
SearchCritEditor.Help	URL	HTML file that is displayed when user selects 'This Screen' from the help menu.

Property Name	*Value Type*	*Description*
Events.height	Integer	Preferred height (in pixels) for the Events panel running on this machine. Set blank to revert to default.
Events.width	Integer	Preferred width (in pixels) for the Events panel running on this machine. Set blank to revert to default.
Events.x	Integer	Preferred horizontal position (X coordinate) for the Events panel running on this machine. Set blank to revert to default.
Events.y	Integer	Preferred vertical position (Y coordinate) for the Events panel running on this machine. Set blank to revert to default.
Events.BufferSize	Integer	The number of events that can be displayed before the oldest events are deleted.
Events.UpdatePeriod	Integer	Number of seconds between automatic updates to the events display. Set to 0 to disable automatic updates. (default=5)
Events.Backlog	Integer	The number of historical events to display when connecting to the LRGS. (default=20).
Event.Help	URL	HTML file that is displayed when user selects 'This Screen' from the help menu.
MessageBrowser.x	Integer	Preferred horizontal position (X coordinate) for the MessageBrowser panel running on this machine. Set blank to revert to default.
MessageBrowser.y	Integer	Preferred vertical position (Y coordinate) for the MessageBrowser panel running on this machine. Set blank to revert to default.
MessageBrowser.height	Integer	Preferred height (in pixels) for the MessageBrowser panel running on this machine. Set blank to revert to default.
MessageBrowser.width	Integer	Preferred width (in pixels) for the MessageBrowser panel running on this machine. Set blank to revert to default.
MessageBrowser.Help	URL	HTML file that is displayed when user selects 'This Screen' from the help menu.
MessageBrowser.Prefix	String	Initial setting of the Prefix string in the MessageBrowser display.
MessageBrowser.Suffix	String	Initial setting of the Suffix string in the MessageBrowser display.

Property Name	*Value Type*	*Description*
MessageBrowser.User	String	Initial setting of the user name field in the Message Browser display.
MessageBrowser.Port	Integer	Default port number for Message Browser display.
MessageBrowser.Timeout	Integer	Number of seconds that browser will wait for a DCP message from the server. (default=5)
MessageBrowser.DefaultSearchCrit	String	Name of default search criteria file. Filename field is initially populated with this name.
MessageBrowser.WrapLongLines	true/false	This controls the initial setting of the Wrap Long Lines checkbox.
MessageBrowser.ConnectionsFile	Filename	Name of the properties file that stores successful server connections. The LRGS name (or IP address), port number, and date of last connection are stored.
MessageOutput.height	Integer	Preferred height (in pixels) for the MessageOutput panel running on this machine. Set blank to revert to default.
MessageOutput.width	Integer	Preferred width (in pixels) for the MessageOutput panel running on this machine. Set blank to revert to default.
MessageOutput.x	Integer	Preferred horizontal position (X coordinate) for the MessageOutput panel running on this machine. Set blank to revert to default.
MessageOutput.y	Integer	Preferred vertical position (Y coordinate) for the MessageOutput panel running on this machine. Set blank to revert to default.
MessageOutput.OutputFile	String	Name of default output file when “Save to File” is selected in the message browser.
MessageOutput.FileExists	String	Three possible values: ‘Fail’ means to disallow writing if the specified output file exists. ‘Append’ means to append to the file if it exists. ‘Overwrite’ will overwrite the file if it exists.
MessageOutput.Timeout	Integer	Number of seconds that program will wait for a DCP message from the server. (default=5)
MessageOutput.CloseWhenDone	true/false	If set to ‘true’, the output window will be automatically closed when output has finished (i.e. when the ‘until’ time is reached).

Property Name	*Value Type*	*Description*
MessageBrowser.ConnectionsFile	Filename	(Default=LddsConnections) This is the file where the browser remembers the host name and port number of recent sessions.
MessageBrowser.PresentationGroup	name	The name of the DECODES presentation group to use for formatting data.
MessageBrowser.DecodesPropFile	Filename	DECODES properties file, which specifies database types, locations, etc.
MessageBrowser.AfterData	String	Initial setting of the "After Data" field.
MessageBrowser.Decode	true/false	Initial setting of the "Decode Messages" checkbox.
MessageBrowser.DefaultSearchCrit	Filename	(Default=MessageBrowser.sc) The search-criteria dialog is initially populated with the contents of this file.
MessageBrowser.UseEditDb	true/false	This flag tells DECODES to use the editable database to decode data.
MessageBrowser.TimeZone	name	(Default=UTC) Time stamps of decoded data are presented in this time zone.
MessageBrowser.OutputFormat	name	The name of the output format for displaying decoded data (default = human-readable).

By selecting File-Properties from the menu of any of the GUI screens, you will be presented with a scrolling list of properties for that screen. An example is shown in Figure 7-2. The two top-level screens (*LRGS Access* and *LRGS Services*) will show all properties. You can modify the property values in the list and press the *Save* button to overwrite the previous properties file.

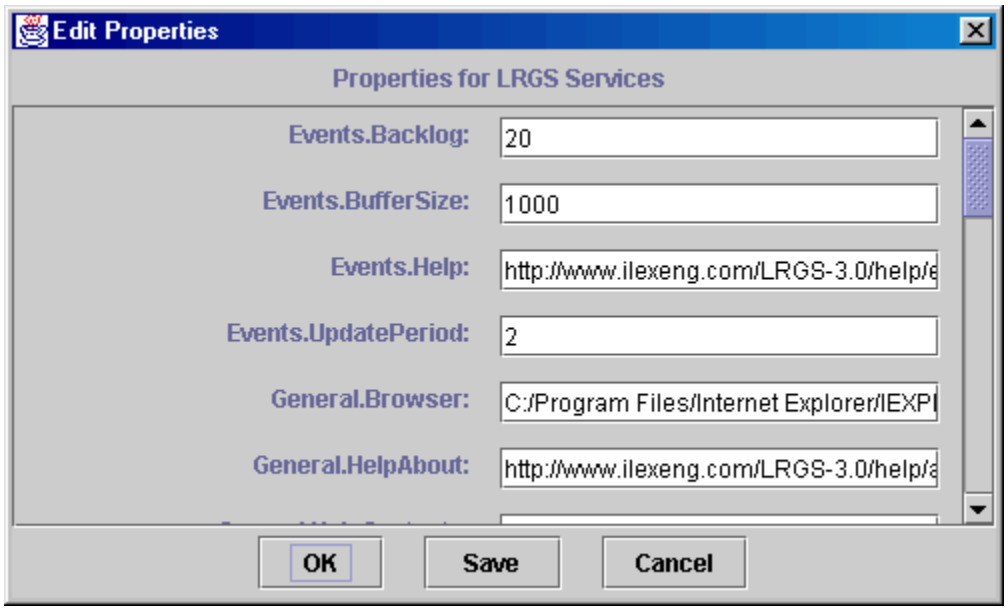


Figure 7-2: Edit Properties Screen. *From this screen you can modify and save the properties that control the appearance and behavior of the GUI.*

Property changes made in this menu take effect immediately. Note that unless you press "Save" the settings will revert to the previous value the next time you start the software.

7.4 On-Line HTML Help Files

Each GUI screen has a help menu with three options:

- Help Contents
- Help on This Screen
- About This LRGS Version

A standard browser is used to display the help files. By default, the help files referenced in the software will be stored on the Ilex Engineering web site. You can change this by modifying the properties, as described in the preceding section.

These on-line manual pages explain the operation of each GUI screen in detail. They are not repeated in this manual. For easy access to these documents, go to:

http://www.ilexengineering.com/LRGS-3.3/help/contents.html

The LRGS installation will also place these HTML help files in /usr/local/lrgs3.2.0/help. Display the file "contents.html" in this directory to gain access to all help pages.

7.5 Starting the Individual LRGS GUI Programs

The LRGS GUI is made up of several Java classes. In general, each screen is implemented as a separate Java class. You can start each screen individually by typing the appropriate command-line. More commonly, however, you will start up the top level "LrgsAccess" screen which contains links to all other screens.

7.5.1 Command Line Arguments Common to all Screens

To connect to a given system, the GUI program must have one of two things:

- URL pointing directly to the LrgsSystem object. The LrgsSystem object provides access to all of the services on a particular LRGS.
- URL pointing to an LrgsDirectory object and a hostname that is advertised in the directory.

When the LRGS is installed, the object-reference files are placed in the web server's directory. So for a given *hostname* (including the domain portion if necessary), the URL would probably be:

`http://`*`hostname`*`/LrgsSystem.ior` (LrgsSystem object)
`http://`*`hostname`*`/LrgsDirectory.ior` (LrgsDirectory object)

You can specify the LrgsSystem URL directly with the –U argument. For example, the following command would start the events-display GUI for the NOAA Wallops system:

```
java lrgs.gui.EventDisplay –U http://cdadata.wcda.noaa.gov/LrgsSystem.ior
```

You can specify the Directory URL by setting the "General.DirectoryIOR" property in the lrgsgui.properties file (See section 7.3), or with the –D argument. If you are starting any GUI but the LRGS Access screen, you will also need to specify a host name. For example, the following command starts the Event Display and looks up a system called 'bloom' in the directory at the Wallops CDA system. (The entire command would be typed on a single line):

```
java lrgs.gui.EventDisplay
   -D http://cdadata.wcda.noaa.gov/LrgsDirectory.ior -H bloom
```

When starting the LrgsAccess GUI, you only need to provide the directory URL. This is because the Access GUI gives you a menu of registered systems from which to pick.

In summary, the following command line arguments are accepted:

- -U *URL* URL containing IOR to LrgsSystem object.
- -D *URL* URL containing IOR or LrgsDirectory object.
- -H *hostname* Works with the –D argument. Look for the specified host name in the directory.
- -P *propertiesfile* Causes the GUI to load properties from the named file, rather than the default location.

7.5.2 List of Individual Screen Classes

The following is a list of LRGS GUI Classes and the command lines that would be used for starting them:

LRGS Access (Select from a menu of available LRGS systems)

```
java lrgs.gui.LrgsAccess
   -D URL
```

… or specify Directory URL in General.DirectoryIOR property.

LRGS Services (Select from a menu of services available on a connected LRGS)

```
java lrgs.gui.LrgsServices
   -U LrgsSystemURL
         -or-
   -D LrgsDirectoryURL -H hostname
```

… Directory URL may also be specified in General.DirectoryIOR property.

LRGS Events Display

```
java lrgs.gui.EventDisplay
   -U LrgsSystemURL
         -or-
   -D LrgsDirectoryURL -H hostname
```

… Directory URL may also be specified in General.DirectoryIOR property.

LRGS Real-Time Status Display

```
java lrgs.gui.RealTimeStatus
   -U LrgsSystemURL
         -or-
   -D LrgsDirectoryURL -H hostname
```

… Directory URL may also be specified in General.DirectoryIOR property.

LRGS Message Browser

```
java lrgs.gui.MessageBrowser -H host -p port
```

… where *host* is the complete host (and domain if necessary) name and *port* is the port number on which the Data Server is listening.

7.6 The GUI Screens

Detailed information on the GUI screens can be found in the on-line reference manual pages at http://www.ilexeng.com/LRGS-3.1/help/contents.html.

7.6.1 The LRGS Selection Screen

CD to the LrgsClient directory and type:

```
lrgsgui      (on Windows)
```

or

```
lrgsgui.sh   (on Unix)
```

A screen will appear as shown in Figure 7-3. By pressing the "Lookup Name" button you will see a list of LRGS systems that have advertised their services, ash shown in Figure 7-4. Alternately, if you know the hostname or IP address of the LRGS you wish to access, you can type it directly in the space provided.

After selecting a system, press "Connect". This will connect you to a specific LRGS and take you to the LRGS Service Selection tool.

Figure 7-3: Top-Level LRGS GUI Screen.

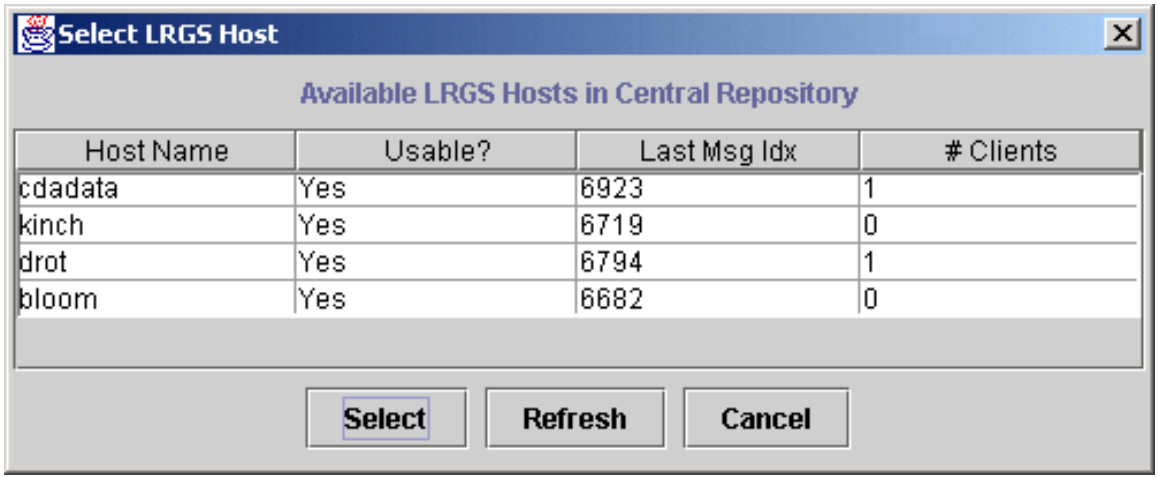


Figure 7-4: Select from available LRGS systems.

7.6.2 The LRGS Services Screen

You can start the LRGS Services Screen by pressing the "Connect" button from the LRGS Selection Screen.

You can also start it directly by typing:

`drotgui` *(on Windows)*

or

`drotgui.sh` *(on Unix)*

Starting the screen this way will connect you directly to the CDADATA machine at Wallops.

The LRGS Services Screen is shown in Figure 7-5. The name of the connected system is shown in the title-bar. The screen shows the overall system status and time. From here you can access other screens by pressing the appropriate button.

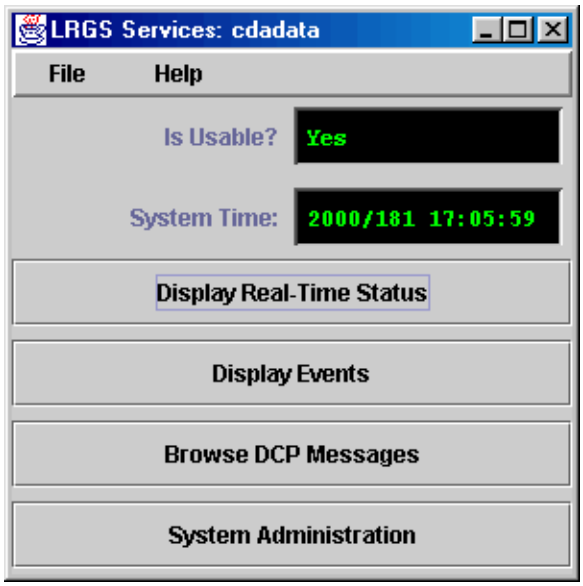


Figure 7-5: LRGS Services Screen.

7.6.3 The Message Browser Screen

You can start the message browser by clicking the "Browse DCP Messages" button from the LRGS Services screen. You can also start it directly by typing:

You can also start it directly by typing:

`msgaccess` *(on Windows)*

or

`msgaccess.sh` *(on Unix)*

The Message Browser screen is shown in Figure 7-6.

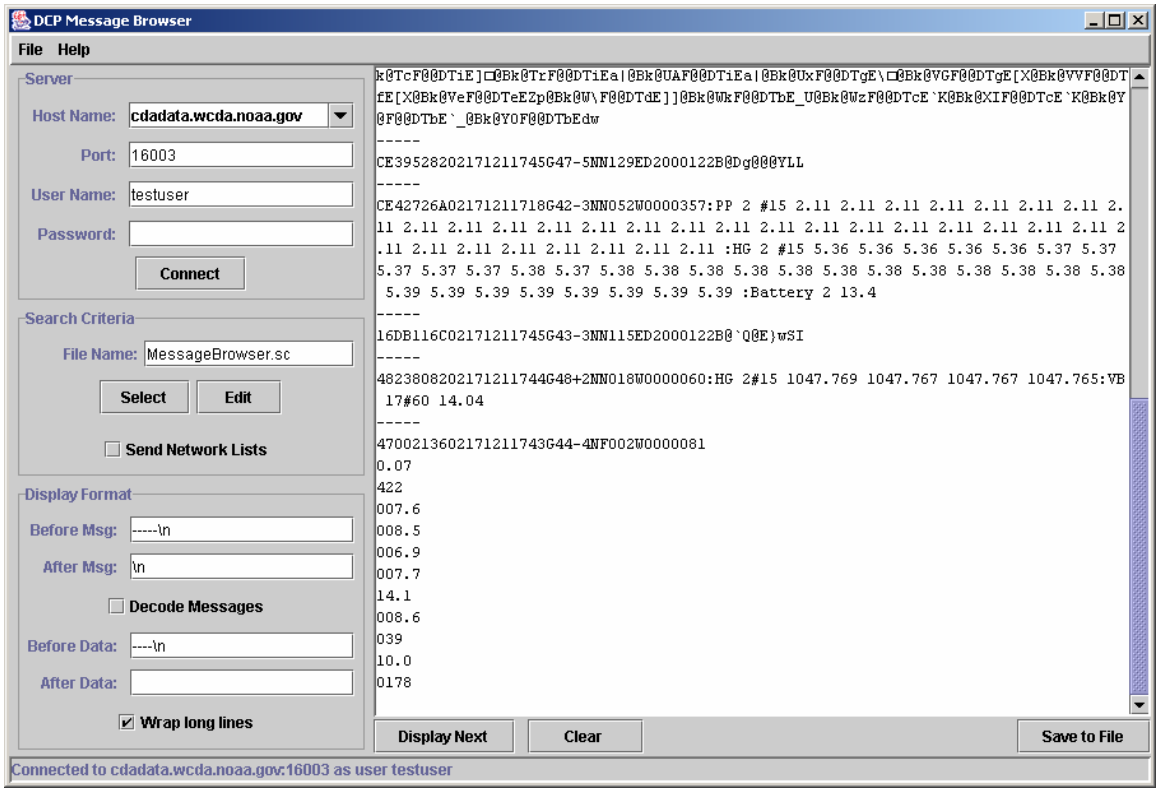


Figure 7-6: Message Browser Screen.

In the upper left quadrant you specify connection information:

- Host Name is either a fully-qualified domain name, an alias, or an IP address specifying the LRGS host you want to connect to.
- Port is a numeric TCP port number. The LRGS DCP data server uses port 17000 by default.
- User Name specifies your ID for connecting to the host.
- The Password field is for authenticated connections to a DDS server.

Leave the password field blank for an unauthenticated connection. *Versions of DDS before LRGS release 3.3 only support unauthenticated connections.*

LRGS version 3.3 (and higher) support authenticated connections to the DDS (DCP Data Service). The DDS may be configured to accept and/or require authenticated connections. Type in a password to cause the Message Browser to attempt an authenticated connection.

The middle-left section of the screen is where you specify search criteria. You can specify the name of a search criteria file. This file will be downloaded when you first try to display a message.

The "Select" button brings up a file-selection dialog for you to navigate to, and select a file. Once selected, you can press the "Edit" button to bring up the Search Criteria Editor screen, as shown in Figure 7-7.

The check-box labeled "Send Network Lists" allows you to specify how network lists are handled. Recall that a search criteria file can specify network lists to be used. These lists might already reside on the LRGS in your user directory, or in one of the LRGS directories. If this is the case, leave this box un-checked.

Conversely, you may be using a new network list that only resides on your client machine. If this is the case, check the box. The network lists will be downloaded to the LRGS before the search criteria file is transferred.

When the server encounters a network list name in a search criteria file, it looks in directories in the following order:

- If a complete path is specified, only that directory is searched.
- The current DCP Data User's directory (This is typically a sub-directory under ~lrgs/users).
- The current user's UNIX home directory, if one exists.
- ~lrgs/netlist
- ~lrgs/netlist/remote
- ~lrgs

The lower-left area controls the Display Format for each DCP message. The 'Prefix' string is printed before the message. The 'Suffix' string is printed after the message. When the 'Wrap Long Lines' check box is checked, the horizontal scroll-bar will disappear. Long lines will be wrapped. When un-checked, lines of data will not be wrapped. Rather, a horizontal scroll bar will appear allowing you to view the entire message.

The 'Before Data' string is printed after the message but before the decoded data. The 'After Data' string is printed after the decoded data.

Check the “Decode Message” checkbox to have the browser attempt to decode the message and display it in engineering units. You must have DECODES installed on your system to use this option. If decoding is successful, you will see the decoded data. If not, you will see an error message explaining the problem.

A common problem is that the platform you are viewing is not in your DECODES database.

Several of the MessageBrowser properties affect how decoding is done. See the list of properties in the table in section 7.3.

7.6.4 The Search Criteria Editor Screen

The search criteria editor screen is shown in Figure 7-7. There are several valid formats for entering timer ranges, as explained in section 7.7. A commonly used technique is to specify times relative to "now", as shown in the figure.

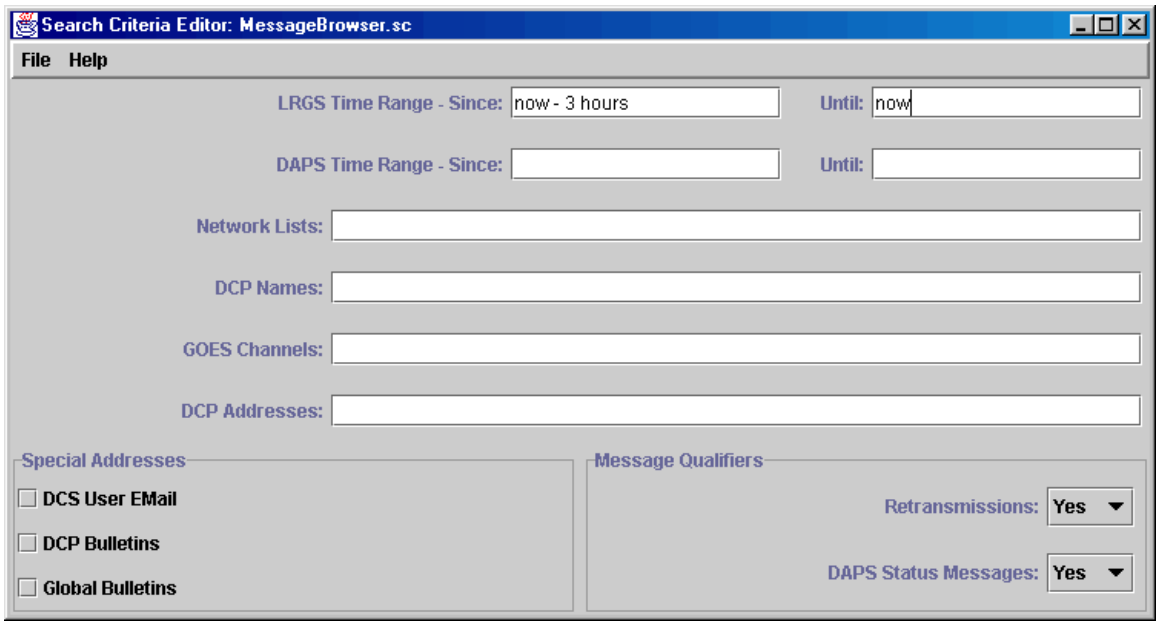


Figure 7-7: Search Criteria Editor.

7.6.5 The Message Output Screen

If you press “Save To File” from the Message Browser screen, The Message Output Screen is displayed, as depicted in Figure 7-8.

In this window you specify an output file to receive the data specified by your search criteria. The radio buttons along the left allow you to specify what to do if the file already exists.

Press 'Run' to start saving data to the file. The DCP address, time-stamp, and message count will be displayed in the screen along the right.

You can pause output by pressing the 'Pause' button. Press 'Run' to continue.

If you want the window to automatically close when the specified 'UNTIL' time is reached, check the box labeled 'Close When Done'.

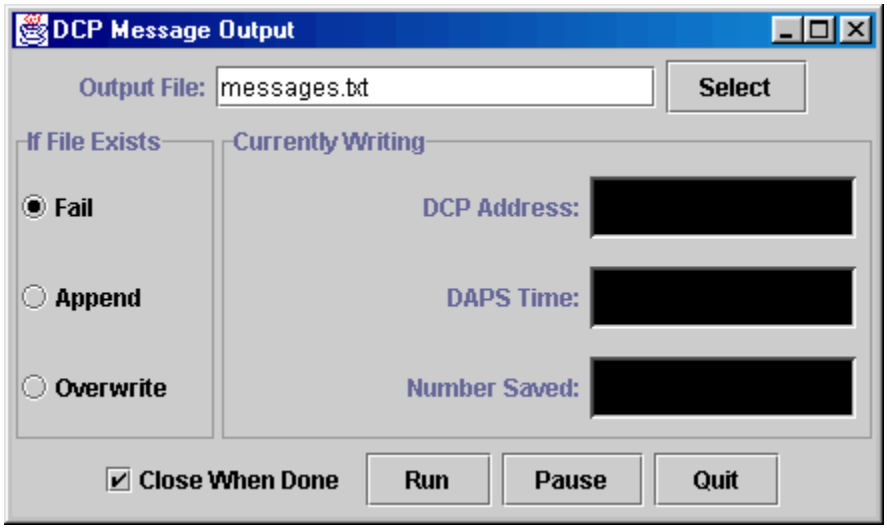


Figure 7-8: The Message Output Screen.

7.7 Search Criteria File Format

A search criteria file is a text file containing a series of keyword-value pairs, one per line. By convention, search criteria files should have a ".sc" extension. Each keyword signifies a particular criterion that DCP messages must pass in order to be returned.

Each line begins with a keyword, followed by a colon, followed by a string value. Here are the available keywords:

DRS_SINCE	Only retrieve messages that were received after the specified time. *See allowable time formats below.*
DRS_UNTIL	Only retrieve messages that were received before the specified time. *See allowable time formats below.*
DAPS_SINCE	Only retrieve messages with a DAPS time-stamp after the specified time. *See allowable time formats below.*
DAPS_UNTIL	Only retrieve messages with a DAPS time-stamp before the specified time. *See allowable time formats below.*
NETWORK_LIST	The value following this keyword is a network list file. Only retrieve messages whose DCP address is contained in the list. For multiple lists, put multiple lines in the search criteria file, each beginning with the NETWORK_LIST keyword.
DCP_ADDRESS	Only retrieve messages with the specified DCP address. To specify multiple addresses, put multiple lines in the search criteria file, each beginning with the DCP_ADDRESS keyword.
DCP_NAME	Only retrieve messages with the specified DCP name. Names are mapped to DCP addresses in network list files. See the section below on Network List Files for details.
CHANNEL	Only retrieve messages that were transmitted on the specified GOES channel. The value is a number only. The GOES spacecraft identifier ('E' or 'W') is not necessary.
SOURCE	Specifies that the client only wants to retrieve data that was received from the named source. Possible arguments are DOMSAT, NETBACK, DRGS, NOAAPORT, LRIT, or OTHER. For multiple sources, put multiple lines starting with 'SOURCE:', each with a single argument.

7.7.1 Allowable Time Formats for a Search Criteria File

The SINCE and UNTIL values can take one several time formats.

Relative formats start with the keyword "now" and then add or subtract increments. For example:

```
now - 20 minutes
now - 1 day
now - 1 week 3 days 20 minutes 10 seconds
now
```

You can specify an absolute GMT value in one of the following formats.

`YYYY/DDD HH:MM:SS`	complete specification
`YYYY/DDD HH:MM`	seconds assumed to be 00
`DDD HH:MM:SS`	assume current year
`DDD HH:MM`	seconds assumed to be 00
`HH:MM:SS`	assume current day
`HH:MM`	seconds assumed to be 00

You can specify that output should start with the last message you retrieved from a previous session. This is a special value that can only appear in the LRGS_SINCE field. Simply type the word:

```
last
```

The "last" keyword provides an easy way to connect periodically and processes all messages that have arrived since your last session. Simply connect periodically and use the time range:

```
LRGS_SINCE: last
LRGS_UNTIL: now
```

The server tracks the last message received by each user. So if you plan to use "last", make sure that no one else is using your DDS account.

7.8 Network List File Format

Network List Files are ASCII text files that contain a series of DCP addresses, one per line. By convention, they should have a ".nl" extension.

The only mandatory restriction on the format of the file is that each line should begin with a hex DCP address (8 chars long). However, several utilities in the LRGS software suite can accept an enhanced format that allows you to associate names and comments with each DCP address:

```
Address:Name Comment
```

- The line should begin with the hex DCP address followed by a single colon.
- The first blank-delimited word following the colon is taken to be the DCP name.
- Any additional text following the name is a free-form comment.

For example:

```
CE123456:BLUE_RIV Blue river at west fork - stage, temp
```

- The DCP address is CE123456
- The DCP Name is "BLUE_RIV". This name can be used for a variety of purposes within the LRGS.

- The comment is “Blue river at west fork – stage, temp”

7.9 Administrative Functions in the LRGS GUI

When you press the “System Administration” on the LRGS Services menu, you will be prompted to enter a user-name and password.

Review section 3.8. That section describes how you create administrative users, enter passwords, and assign the “admin” privilege. Only usernames entered in this manner will be able to use the administrative functions of the LRGS GUI. *Note: These are NOT Unix login accounts.*

If you enter a proper username and password you will see an extension of the services menu as shown in Figure 7-9.

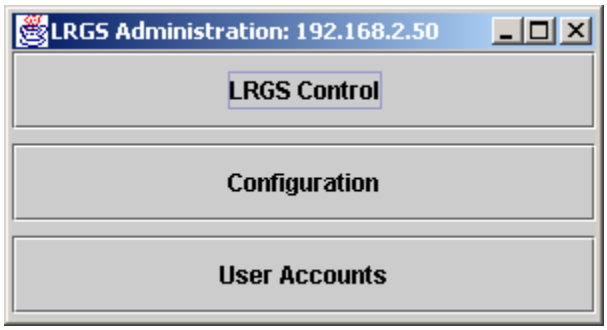


Figure 7-9: Administrative Services Menu.

7.9.1 The LRGS Control Screen

The LRGS Control Screen is shown in Figure 7-10.

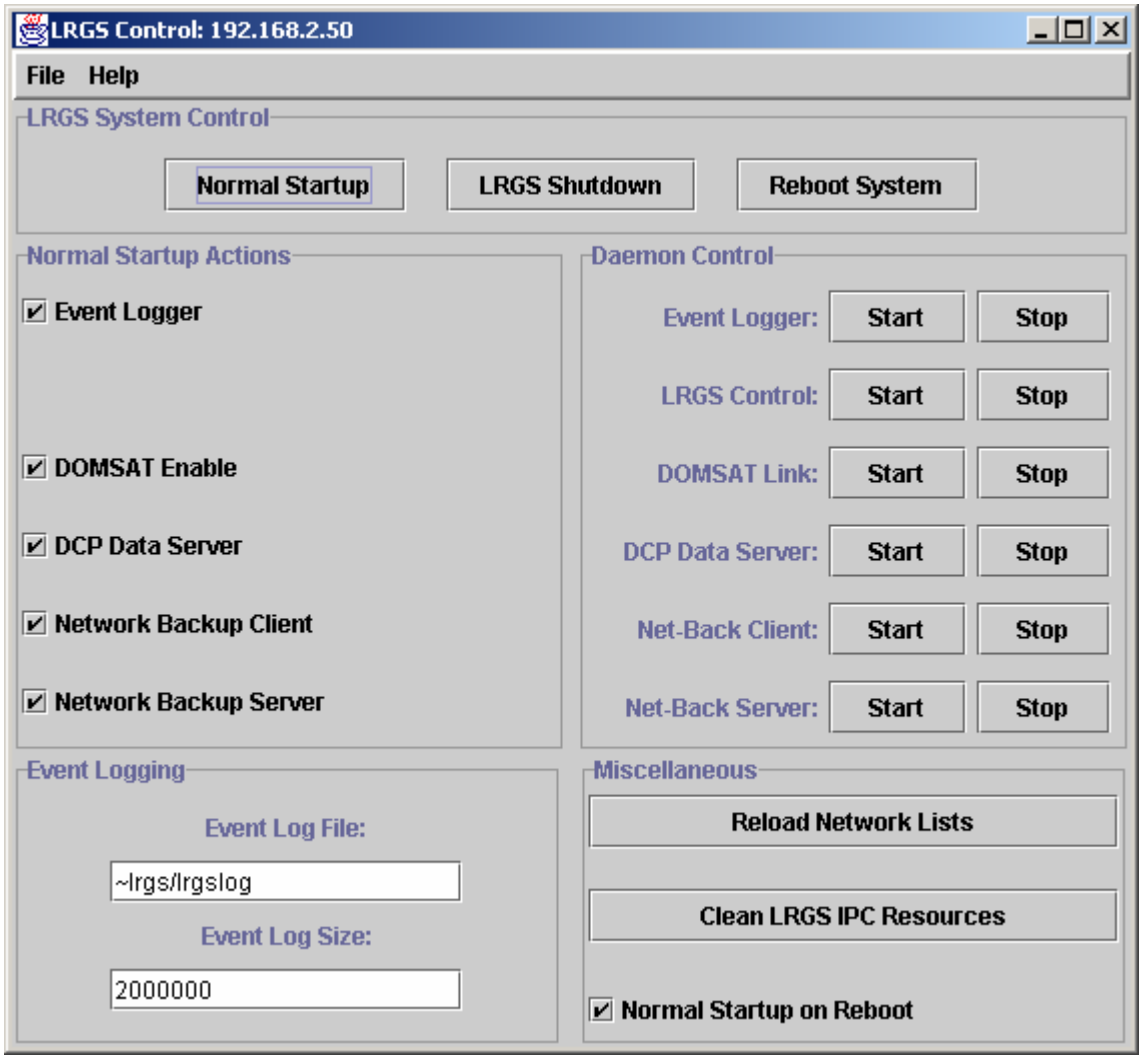


Figure 7-10: LRGS Control Screen.

There are three buttons along the top of the screen:

- **Normal Startup:** Starts the LRGS Software. If the software is already running, this will have no effect
- **LRGS Shutdown:** Stops the LRGS Software. If the software is not currently running, this will have no effect.
- **Reboot System:** This button will cause the LRGS system to reboot. (Note. User 'lrgs' must be granted root privileges in order for this to work.)

The area labeled "Normal Startup Actions" defines what happens when the system performs a "normal" startup. In most cases you will want all boxes checked.

The area labeled "Daemon Control" allows you to start and stop individual "daemon" processes on the LRGS. This is mostly used in trouble-shooting scenarios.

The area labeled “Event Logging” defines the location and size of the event log file. The size that you enter is the maximum number of bytes that the file will be allowed to grow to. After this, the file will be renamed with a “.old” extension, and a new log file will be opened with the specified name.

The area labeled “Miscellaneous” contains three controls:

- The “Reload Network Lists” button causes all network lists to be immediately reloaded. Use this in trouble shooting situations after you have modified a network list. Normally, when a network list is modified, it is automatically reloaded within 10 minutes.
- The “Clean IPC Resources” button should not be used except by developers.
- The “Normal Startup on Reboot” checkbox should normally be checked. This causes a normal startup to occur when your LRGS system is rebooted for any reason.

7.9.2 The LRGS Configuration Screen

In the current release of LRGS, configuration files must be edited manually. That is, with a generic text editor like gedit or vi. However, the GUI does provide methods for retrieving, installing, and activating new configuration files.

The LRGS Configuration menu is shown in Figure 7-11. The top area of the menu allows you to select and edit local copies of configuration files.

The bottom area contains three buttons for controlling the LRGS you are connected to:

- **Install** means to install the specified file as the “lrgsconfig” on this LRGS. Once installed, this configuration will take effect whenever the LRGS is restarted, or the “Activate” button is pressed.
- **Retrieve** retrieves the lrgsconfig from the LRGS you are connected to and saves it in a local file on your disk. You will be prompted for a file name.
- **Activeate** causes the LRGS to reload its configuration. Any changes will then take effect.

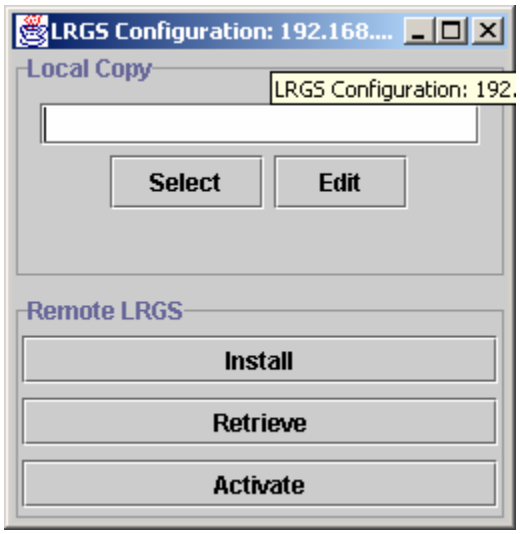


Figure 7-11: LRGS Configuration Dialog.

7.9.3 DDS User Accounts

You can add and delete DDS (DCP Data Service) user accounts through the administrative GUI. The menu is shown in Figure 7-12.

When you press **Add** you are prompted for the new user name, and optionally, a password. The dialog is shown in Figure 7-13. There are two type-in fields: user name, and password. Leave the password field blank to create an "unauthenticated" user account. Enter a password for an "authenticated" user account. The middle part of the screen gives you information about the DDS server on the LRGS you are connected to. There are three possibilities:

- The server is running LRGS software before release 3.3 and therefore does not support password-authentication.
- The server is release 3.3 or above and supports either authenticated or non-authenticated users.
- The server is release 3.3 and has been configured to require authentication. In this case, you MUST enter a password for all users.

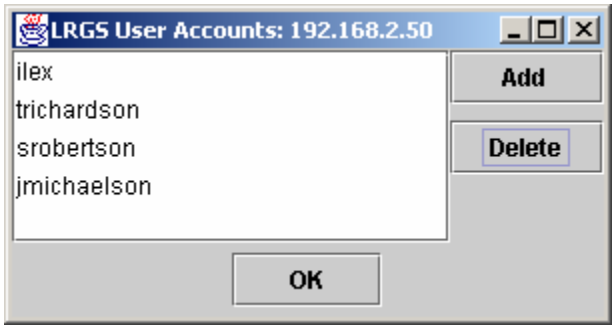


Figure 7-12: DDS User Accounts Dialog.

Figure 7-13: Add DDS User Dialog.

8. LRGS Configuration

The LRGS Configuration is stored in the file ~lrgs/lrgsconfig. After a fresh installation, this will be a symbolic link to a file in the installation directory under /usr/local.

8.1 Customizing a Local Copy of the File

To customize your configuration, first make this a real file, not a symbolic link. Login as 'lrgs'. Then in your home directory type:

```
cp lrgsconfig tmp
rm lrgsconfig
mv tmp lrgsconfig
```

Doing this will prevent future upgrade installations from overwriting your custom configuration settings.

8.2 Configuration File Format

The file is an ASCII file that you can modify with any standard text editor.

Lines that begin with a '#' character are considered comments.

Each line contains a keyword, a colon character ':', and a value. The entire value must be contained in a single line. Continuation lines are not supported.

The following list describes each keyword, and what kind of value it can take.

Keyword	*Value Type*	*Description*
NETLIST_FILTER	On/Off	Controls whether incoming messages will be filtered by network list.
DUP_FILTER	On/Off	Controls whether incoming messages will be filtered to remove duplicates.
MESSAGEFILE	Filename	The path to the DCP message storage file. The LRGS will created this file along with an index file with a '.I' extension.
MAX_STORAGE	Number of Bytes	Maximum size (in bytes) for the message file. Once this size is reached, the oldest messages will be over-written as new messages arrive.
MAX_RECORDS	Number of Records	The maximum number of DCP messages that can be stored in the message file. Once this number is reached, the oldest messages will be over-written as new messages arrive.
DOMSAT_TIMEOUT	Number of seconds	A timeout is declared after this many seconds if no messages are received over the DOMSAT link.
ICP_DEVICE	Filename	The name of the Unix device node for the Franklin ICP188 card. This is only necessary if your system uses this board.
ICP_PROGRAM	Filename	The name of the file containing the embedded code to be loaded onto the Franklin ICP188 board. There are two possibilities: ~lrgs/bin/188x25.exe is the code for a live DOMSAT link. ~lrgs/bin/domsim.exe is a simulator that generates dummy messages.

NETLIST	Filename + Y/N	A file name of a local network list. If a 'Y' is placed after the file name (separated by a space) then DCP names will be supported from this list. A 'Local' list is one that is used only on this LRGS.
SHARED_NETLIST	Filename + Y/N	A file name of a shared network list. If a 'Y' is placed after the file name (separated by a space) then DCP names will be supported from this list. A 'Shared' list is made available to other systems by the distributed CORBA API.
REMOTE_NETLIST	host:Filename + Y/N	Specifies a remote network list. Remote lists are downloaded from the specified host via the distributed CORBA API. Local copies are placed in ~lrgs/netlist/remote. If a 'Y' is placed after the file name (separated by a space) then DCP names will be supported from this list. Note 'host' can be a simple host name, as advertised in this LRGS' directory, or a complete domain/host name for direct connection.
BER_THRESHOLD	two digits	Bit Error Rate Threshold: The first digit is a multiplier. The second digit is a negative power of 10. For example "18" means 1 X 10^-8.
TIMEOUT_PROG	Command line	This command is executed when a DOMSAT Timeout occurs.
RECOVER_PROG	Command line	This command is executed when the DOMSAT link recovers from a timeout situation.
BER_ALARM_PROG	Command line	This command is executed when the Bit Error Rate Threshold is exceeded.
KID	Command line	This command is a daemon that is executed in the background when the LRGS is started. A specific protocol is required between these daemons and the LRGS control process. You cannot add arbitrary commands.
COMSTREAM_PORT	Filename	The name of the serial port device node to be used for communication with the ComStream M&C port
COMSTREAM_BAUD	Integer	The baud rate to use on the ComStream serial port. Unless you reconfigure the ComStream unit, this must be 1200.
COMSTREAM_DOWNLINK	String	The name of the daemon that is handling the DOMSAT downlink. This will be one of the daemons specified with a 'KID' line.
NETBACK_TIMEOUT	Number of seconds	Maximum seconds that the network backup client will wait for a response from a server. 60 is a reasonable value.
NETBACK_FUDGE_TIME	Number of seconds	Network backup correlates messages by sequence number and approximate receipt time. This is the maximum variance for receipt time. 600 (ten minutes) is a reasonable value.
NETBACK_SITE	host:port	This tells the network backup client how to connect to a host. There can be several NETBACK_SITE lines. 'host' is either a host name or an IP address. 'port' is the port number on which the network backup server is listening on that host.
TCPIP_CMD:	Command line	This command line is used to start the LRGS DCP Data Server when it is enabled.
NETBACK_CLI_CMD	Command line	This command line is used to start the Network Backup Client when it is enabled.
NETBACK_SVR_CMD	Command line	This command line is used to start the Network Backup Server when it is enabled.

DCP_DATA_USER_DIR	Directory name	This directory will hold sub-directories for each of the Data Server clients registered for this machine. Inside a users sub-directory will be placed an search criteria, or network list files that the user downloads.
DDS_REQUIRE_AUTH	true/false	Set this to true to require that DDS (DCP Data Service) users authenticate themselves before gaining access to your system. The default value is "false" if the property is not present. This means that un-authenticated users will be accepted.
DDS_AUTH_MECHANISM	*name*	The name of the mechanism to be used by DDS to authenticate incoming users. Currently the only supported mechanism is "password". This is also the default value used if the property is not present.
STARTUP_TIMEOUT	Number of seconds	When starting LRGS, this is the time that the control process will wait for all daemons to finish initializing. You may need to increase this value if you have a very large message file and/or a slow hard disk. In such cases the lrgs_archive_d process may take several minutes to initialize.
MERGE_DURATION	Number of seconds	The DRGS merge filter uses this time value to match messages from different sources. Messages from same DCP on same channel received within this many seconds are to be considered copies of the same message. Default is 300 seconds (5 minutes).
MERGE_PREFERRED_SRC	Name	This should be "DOMSAT", "DRGS", or a downlink number. The DRGS merge filter uses this. If two copies of the same messages are received, and both are of questionable quality; THEN keep the one from the preferred source.
MERGE_FILTER_ENABLE	true/false	Enables the DRGS merge filter (default = true).

8.3 Child Processes in the Configuration File

There are several lines in the configuration file that begin with the prefix "KID:". These lines specify the child-processes that will be managed as part of the LRGS suite. In some cases, you will configure these daemon processes with command line arguments.

All of the KID processes will honor the following arguments:

`-n` *`procname`* — Sets the name that this process uses when reporting events in the log file. It is normally not necessary to use this argument, because each process has an appropriate default name.

`-d` *`debuglevel`* — The debug level is 0 (default=least verbose) through 3 (most verbose). Debug messages are sent to the LRGS log file and are also visible with the 'showevents' utility.

Some KID process also support custom arguments.

8.3.1 ComStream Daemon

The lrgs_comstream_d process polls the M&C port of the ComStream receiver periodically. It can take the following arguments:

`-pn, -pe, -po` — Sets parity for the serial port to none, even, or odd respectively. The default is even.

-b7, -b8	Sets the number of bits per character to 7 or 8 respectively. The default is 7.

8.3.2 Downlink Daemon Processes

Downlink processes include lrgs_sangoma_d (the most common), lrgs_icp188_d, and lrgs_wanic_d. These processes can also take a '-p' argument that indicates that this is the primary downlink. This distinction is used by the network-backup mechanism: Only data dropped by the primary downlink will be recovered.

8.3.3 Network Backup Server

The Network Backup Server is a Java program. It is started as follows:

```
java lrgs.lnb.LegacyNetbakSvr  args...
```

… where args can be:

`-p portnum`	The server will listen for connections on this port number. The default is 17001.
`-b IPaddress`	Bind the listening socket to a particular network interface. Use this option if your server has multiple NICs and you only want it to listen on a particular network.
`-d debuglevel`	The debug level is 0 (default=least verbose) through 3 (most verbose). Debug messages are sent to the LRGS log file and are also visible with the 'showevents' utility.
`-A`	This option means 'Allow Bad Primary'. It defaults to false. Use this option if you want to server data to clients even if your primary downlink is inactive.
`-u usage-log`	The server records all activity in a usage log with default name "~/netback-log". This option allows you to specify a different name.

The server writes to a usage-log which is allowed to grow to a length of 5 megabytes. When this occurs it is renamed to ~/netback-log.old and a new log is opened. The log contains text lines in the following format:

```
time-stamp  clientNum  clientName  #messages  lastActivity
```

…where:

- *time-stamp* (YYYY-MM-DD/HH:MM:SS) is the time that the log messages was recorded.
- *clientNum* is a unique number assigned to this client when the connection was made. It is only unique to a particular run of the server.
- *clientName* in the format *user@hostname*.
- *#messages* is the number of messages received by the client since the last log message.
- *lastActivity* is the time of the last request of any kind from this client.

Log messages are made once per minute or when a client disconnects.

8.3.4 DCP Data Server (DDS)

The DCP Data Server is a Java program. It is started as follows:

```
java lrgs.ldds.LrgsDcpDataSvr  args...
```

… where args can be:

`-p` *`portnum`*	The server will listen for connections on this port number. The default is 17001.
`-b` *`IPaddress`*	Bind the listening socket to a particular network interface. Use this option if your server has multiple NICs and you only want it to listen on a particular network.
`-d` *`debuglevel`*	The debug level is 0 (default=least verbose) through 3 (most verbose). Debug messages are sent to the LRGS log file and are also visible with the 'showevents' utility.
`-a`	This option means 'Require Password Authentication'. It defaults to false. Use this option if you want to force clients to authenticate via password.
-m *max-clients*	Specifies the maximum number of simultaneous clients to be accepted.
`-u` *`usage-log`*	The server records all activity in a usage log with default name "~/netback-log". This option allows you to specify a different name.

The server writes to a usage-log which is allowed to grow to a length of 5 megabytes. When this occurs it is renamed to ~/dds-log.old and a new log is opened. The log contains text lines. The format of each line is the same as described for the network backup server in section 8.3.3.

8.4 Sample Configuration File

```
ICP_DEVICE: /dev/icp188
NETLIST_FILTER: On

MESSAGEFILE: /usr/lrgs/messages
MAX_STORAGE: 10000000
MAX_RECORDS: 100000
DOMSAT_TIMEOUT: 60
ICP_PROGRAM: ~/bin/188x25.exe

NETLIST: ~lrgs/dnjtrn.nl Y
SHARED_NETLIST: ~lrgs/dohclb.nl Y
SHARED_NETLIST: /usr/lrgs/test/sim_nl3.nl Y
REMOTE_NETLIST: LRGS1:sim_nl4.nl Y
DUP_FILTER: Off
BER_THRESHOLD: 18
TIMEOUT_PROG: ~lrgs/test/DomsatTimeout.sh
BER_ALARM_PROG: ~lrgs/test/BerAlarm.sh
RECOVER_PROG: ~lrgs/test/DomsatRecovery.sh

#KID: lrgs_icp188_d -n lrgs_icp188_d -p
#KID: lrgs_sim_d -n lrgs_sim_d -d3 -p
#KID: lrgs_et_d -n lrgs_et_d -d3 -p
KID: lrgs_sangoma_d -n lrgs_sangoma_d -d3 -p
KID: lrgs_archive_d -n lrgs_archive_d
KID: lrgs_periodic_d -n lrgs_periodic_d -d3
KID: lrgs_comstream_d
COMSTREAM_PORT: /dev/cua0
COMSTREAM_BAUD: 1200
#COMSTREAM_DOWNLINK: lrgs_et_d
#COMSTREAM_DOWNLINK: lrgs_sim_d
COMSTREAM_DOWNLINK: lrgs_icp188_d
NETBACK_TIMEOUT: 60
NETBACK_FUDGE_TIME: 600
NETBACK_SITE: lrgswallops:17001
TCPIP_CMD: java lrgs.ldds.LrgsDcpDataSvr -d2
NETBACK_CLI_CMD: java lrgs.lnb.LegacyNetbakClient -d1
NETBACK_SVR_CMD: java lrgs.lnb.LegacyNetbakSvr -d1

DCP_DATA_USER_DIR: ~lrgs/users
DDS_REQUIRE_AUTH: true
DDS_AUTH_MECHANISM: password
```

8.5 Distributed Network Lists

There are three different ways of specifying network lists in the LRGS configuration file. These are for local lists, remote lists, and shared lists.

In each case, the [Y|N] following the list name means that the list name can optionally be followed with a 'Y' or an 'N' indicating whether or not the list contains DCP names that you want to be able to use on your system.

NETLIST: *filename* [Y|N]

This is compatible with the legacy format for the configuration file. It specifies a locally-stored list that is not shared with other systems.

SHARED_NETLIST: *filename* [Y|N]

Shared lists are stored locally on your hard disk. The LRGS API makes these lists accessible to other systems.

Each shared list is represented by a SharedFileDigest object on your local name service. The name for the object will be the simple filename with any path component removed. For this reason, make sure all of your shared lists have a unique name.

REMOTE_NETLIST: *hostname:filename* [Y|N]

Remote lists are stored on other systems. You must specify the hostname for the remote system as well as the file name. The filename should be the simple filename without any path components.

Copies of remote lists will be downloaded and stored locally in the directory ~lrgs/netlist/remote. A background process will periodically (every 15 minutes) check to see if the list has changed on the remote system, and if so, it will download a new copy.

9. Using LRGS for a Direct GOES Receive Station

LRGS 3.4 has a feature that allows it to support GOES demodulator systems. Any demodulator system that supports the NOAA-published DAMS-NT (Data Acquisition and Monitoring System - New Technology) ICD can supply data to an LRGS. The LRGS supports up to 64 simultaneous DAMS-NT connections.

9.1 Add the DRGS Input Process to your LRGS Configuration

You must modify the 'lrgsconfig' file to start the DRGS daemon process. Do this by adding a line as follows:

```
KID: java lrgs.drgs.DrgsInput -f drgsconf.xml -n DRGS
```

If you want to see verbose debugging messages in the log, you may also add -d1, -d2, or -d3 to the end of the command line (-d3 is the most verbose).

9.2 Configure your DRGS Interfaces

Next you need to edit the file 'drgsconf.xml' found in the LRGS home directory. This is an XML file. A sample file with two DRGS connections is shown below:

```
<?xml version="1.0"?>
<drgsconf>
  <debug>3</debug>
  <connection number="0" host="drgs-e.mydomain.gov">
        <name>EAST-DRGS</name>
        <enabled>true</enabled>
        <msgport>17010</msgport>
        <evtport>17011</evtport>
        <evtenabled>false</evtenabled>
        <startpattern>534D0D0A</startpattern>
  </connection>
  <connection number="1" host="drgs-w.mydomain.gov">
        <name>WEST-DRGS</name>
        <enabled>false</enabled>
        <msgport>17010</msgport>
        <evtport>17011</evtport>
        <evtenabled>false</evtenabled>
        <startpattern>534D0D0A</startpattern>
  </connection>
</drgsconf>
```

If a <debug> record is included, it should contain a single digit 0, 1, 2, or 3. If no record is found, debugging defaults to level 0, meaning that only warnings and failures will generate log messages.

You can include up to 64 "connection" records, each with a unique "number" argument from 0…63. The host argument is required.

As shown in the sample, you can include options inside the connection records:

- <name> specifies the name of this downlink on the real-time status page, and in log messages.
- <enabled> defaults to 'true'. You may set it to false to temporarily disable receiving messages from a particular DRGS.
- <msgport> defaults to 17010 as per the NOAA DAMS-NT ICD. Only change it if your DRGS uses a non-standard port number.
- <evtport> defaults to 17011 as per the NOAA DAMS-NT ICD.
- <evtenabled> defaults to 'true'. Set it to false to disable event reporting from a particular DRGS. Events from the DRGS are turned into LRGS log messages.
- <startpattern> defaults to the value shown in the sample. This is equivalent to the pattern "SM\r\n", SM stands for Start Message. Enter the 8-hex-digits appropriate for your DRGS.

When doing your initial testing, set the debug level to 3 (most verbose). This can be done with an XML record as shown, or with the -d argument on the command line.

After making these configuration changes, restart the LRGS software and monitor the via the command:

```
showevents
```

You can review historical events in the file ~lrgs/lrgslog.

Also run the command

```
showrtstat -u1 -c
```

This will show the DRGS connections.

9.2.1 Changing the Connection Configuration

You can change the connection-configuration on the fly. After modifying the configuration file, send the SIGHUP signal running to the running DRGS daemon. First type:

```
ps -f -u lrgs
```

Look for the first process that has a comman line "java lrgs.drgs.DrgsInput". The Process ID (PID) will be in the second column. For example, if the PID is 12345, type:

```
kill -SIGHUP 12345
```

Be sure to examine the lrgslog file after doing this. Any problems in parsing the configuration file changes will be reflected in log messages.

9.3 Configure the DRGS / DOMSAT Merge Filter

If you have both a DRGS and DOMSAT interface, then you have some control over how messages from the two are merged.

If you want to keep all messages from both interfaces, then turn off the merge filter completely by adding the following to the lrgsconfig file:

```
MERGE_FILTER_ENABLE: false
```

Otherwise, add the following (note: true is the default setting if there is none specified).

```
MERGE_FILTER_ENABLE: true
```

The merge filter looks for copies of the same message appearing on both links. It then tries to save only the best copy. The rules are as follows:

1. Two messages from the same DCP address on the same channel that arrive within MERGE_DURATION seconds are to be considered copies of the same message.
2. The link provides an indication as to whether a message is GOOD, or BAD (i.e. contains parity errors.)
3. If two GOOD copies are received, the one arriving first is archived, the other is discarded.
4. If a one GOOD and one BAD copy is received, the GOOD one will be saved and the BAD one discarded.
5. If two BAD copies are received, then the one from the "Preferred Source" will be kept and the other discarded.

To specify the duration in seconds of the merge filter, add the following to the lrgsconfig file.

```
MERGE_DURATION: #Seconds
```

For example, to set the merge duration to 10 minutes, add:

```
MERGE_DURATION: 600
```

To specify that DRGS is to be considered the preferred source, add:

```
MERGE_PREFERRED_SRC: DRGS
```

To specify that DOMSAT is the preferred source:

```
MERGE_PREFERRED_SRC: DOMSAT
```

9.3.1 Changing the Merge Filter Configuration

In the current implementation, you must restart the LRGS software in order for changes to the merge filter settings to take effect. The easiest way to do this is to type the following commands while logged in as user 'lrgs':

```
lrgs stop
```

Pause for about 30 seconds, then type:

```
lrgsAutoStart
```

9.4 How to Retrieve DRGS Messages from the LRGS

Messages received on a DRGS link are stored in the same circular file along with DOMSAT messages. They are available to all programs that pull messages via DDS (DCP Data Service). This includes the message browser, the showmsg utility, the GetDcpMessages program, and DECODES.

DDS clients can distinguish messages that originated from a DRGS by looking at the 2-byte uplink-carrier status in the DOMSAT header (see section 1.1.4). For DOMSAT messages, this will be filled with 2-hex digits. For DRGS messages this will be the string “DR”, which stands for “direct readout”.

10. Trouble Shooting FAQ

This section addresses problems you may encounter in installing and running the LRGS.

10.1.1.1.1 Q: The LRGS Status Page says that my system is not usable. Why not?

A: An LRGS is considered 'usable' if:

- All of the daemons are up and running
- It's primary downlink is receiving data – or –
- It's hot-backup downlink is receiving data

You can check the downlink status in the scrolling list on the real-time-status screen. If it is 'Error' or 'Timeout' then the downlink has a problem.

If this is your primary DOMSAT downlink, try enabling and disabling it as follows:

- Bring up the LRGS Control GUI (under the System Administration GUI)
- In the daemon control window (along the right), under DOMSAT Link, first press the Stop button, wait a few seconds, then press the Start button.

10.1.1.1.2

10.1.1.1.3 Q: After booting the LRGS, the clock is not adjusted properly for daylight savings time. How do I make this happen automatically.

A: Login as root and run the 'linuxconf' utility. One of the last items in the list is "date & time". In this menu check the check-box that says "Store date in CMOS Universal Format". Now set the clock properly to your local time. When you do the next reboot, the clock will be adjusted automatically.

Q: I can't use the Message Browser – Save-to-File feature when connecting to my legacy DRS system. Why not?

A: Many legacy DRS systems are configured to only allow one connection per user. If you press 'Connect' on the browser window, it establishes a connection. The Message Output window makes its own connection. The work-around is to re-start the Message Browser. Specify the server, port user-name, and search-criteria file, but do not press Connect. Press Save-To-File and output should work.

11. Specific Scenarios

11.1 Saving and Restoring the DCP Message File

For some applications it is important to save the raw DCP messages permanently. The LRGS software provides the following strategy for accomplishing this:

- Size the 'messages' file so that a compressed copy will fit in a single (650MB) CD.
- Determine how long it takes to accumulate this much data.
- Make a copy of the messages file on a CD periodically.
- When the data is needed, restore the files and run the portions of the LRGS software necessary for accessing the file.

The remainder of this section describes these steps in more detail.

11.1.1 Configure the DCP Message File Size

We have experimentally determined that gzip will compress DCP messages with about a 2:1 ratio. Hence you should set the size of the messages file to 1.2 gigabytes, leaving an ample fudge-factor.

- Login as lrgs.
- Edit the file lrgsconfig.
- find the line that starts 'MAX_STORAGE and set it as follows:

```
MAX_STORAGE: 1200000000
```

Also set the MAX_RECORDS value as follows:

```
MAX_RECORDS:   30000000
```

You now have to delete the existing message storage file, and its index:

```
rm messages
rm messages.I
```

You can now restart the LRGS.

11.1.2 Determine the Backup Period

The time it takes to store 1.2 gigabytes of message data will vary depending on how you use network lists. Allow the LRGS to run for a 24 hour period. Then login as lrgs, and type:

```
ls –l messages
```

This will tell you the size of the message file. From this, determine approximately how long it will take to fill up 1.2 GB of storage. Then subtract several days to guarantee an overlap.

11.1.3 Backup the Message File

The file called 'messages' in the lrgs directory is the one you need to back up. You do not need to save the index file (messages.I) because it can be recreated from the messages.

To save raw DCP message, copy it to a temporary location. We also recommend adding a date-stamp to the file name.

For example, suppose it is now December 13, 2001:

```
cp messages /tmp/messages-20011213
```

This can be done while the LRGS is running.

Depending on how you have your hard disk partitioned, there may not be enough space under the /tmp directory. If this is the case, use the unix 'df' command to determine a directory with ample storage.

Next, compress the file with gzip:

```
cd /tmp
gzip messages-20011213
```

This will result in a file called 'messages-20011213.gz' in the /tmp directory which should be less than 650 MB. Use FTP to transfer this file to the machine where your CD-burner is. Be sure to use binary-mode in FTP.

Copy the file onto the CD and place it in storage.

11.1.4 Restore an Old Message File

At this point, accessing the data in the file will require temporarily disabling your LRGS.

(Future versions of the LRGS may include a utility for accessing data in files without disturbing LRGS operations.)

First login as lrgs and shutdown the LRGS software:

```
lrgs stop
```

Save the 'real' message and index files:

```
mv messages messages.sav
mv messages.I messages.I.sav
```

Now we will copy the old messages file into LRGS and uncompress it. Place the CD in the drive, then:

```
mount /dev/cdrom                  (note: this may be done automatically)
cp /mnt/cdrom/messages-20011213.gz messages.gz
gunzip messages.gz
```

You now have the old file in the LRGS directory called 'messages'. Recreate the index for it:

```
makemsgindex messages
```

You can now restart LRGS by typing:

```
lrgs start
lrgs datasvr enable
```

The above command will not start DOMSAT data reception. You probably do not want to start writing new data on top of this old file.

The above command starts the LRGS and the network daemon for serving out DCP messages. You can now retrieve the data of interest.

11.1.5 Putting the LRGS back to Normal

After you have retrieved the old data, you need to restore the 'current' message files. First stop the LRGS software:

```
lrgs stop
```

Then throw away the old message and index file (you still have your zipped copy on CD):

```
rm messages
rm messages.I
```

Restore the 'current' versions of these files:

```
mv messages.sav messages
mv messages.I.sav messages
```

Now do a full LRGS start-up by typing:

```
lrgsAutoStart
```

Appendix A. Glossary

API	Application Program Interface
CGI	Common Gateway Interface – A method of invoking a script or program on a web server.
CORBA	Common Object Request Broker Architecture – provides a platform-independent and language-independent way of invoking remote services.
COTS	Commercial Off-The-Shelf
DAPS	DCS Automated Processing System – This is a large computer system operated by NESDIS in Wallops, VA. A primary function is to receive data from all platforms and rebroadcast via DOMSAT and an NMC X.25 link. NESDIS has plans to renovate DAPS in the near future.
DCS	Data Collection System – This is the name for the large organization of user agencies, NOAA/NESDIS, and vendors that operate and have a vested interest in environmental monitoring via GOES.
DECODES	DEvice COnversion and DElivery System – This is a legacy software package developed by USGS/WRD.
DOMSAT	Domestic Satellite – This is used as a high-speed broadcast of data from all platforms in the DCS.
DROT	DOMSAT Receive Only Terminal – Public domain prototype system developed for NOAA/NESDIS.
DRS	DOMSAT Receive Station – This is a proprietary system for receiving DOMSAT data. Also see LRGS
EMIT	Environmental Message Interpreter Translator – A proprietary software package in wide use within USACE.
GFE	Government Furnished Equipment
GOES	Geostationary Operational Environmental Satellite – In addition to the primary imaging function, the GOES spacecraft also support a multi-channel data relay which is used by the DCS to facilitate remote environmental monitoring.
GUI	Graphical User Interface
HTML	Hypertext Markup Language – the language of web pages
HTTP	Hypertext Transfer Protocol – The protocol used between browsers and web servers.
JDK	Java Development Kit – A suite of tools provided by Sun Microsystems for java development.
LRGS	Local Readout Ground Station – This is a freely available, open-source system for receiving environmental data via DOMSAT.
NESDIS	National Environmental Satellite Data Information Service – This is the division of NOAA that operates the GOES spacecraft and its relay functions.
NMC	National Meteorological Center

NOAA	National Oceanic Atmospheric Administration – An agency under the U.S. Department of Commerce
NWS	National Weather Service
PKI	Public Key Infrastructure
RMI	Remote Method Invocation – a means whereby a Java application can invoke a method in another Java application, possibly on another machine.
SATIN	Satellite Input – A USGS application for ingesting DCP data into a database.
SSL	Secure Socket Layer – A method of encryption and authentication employed by secure web services.
STIWG	Satellite Telemetry Interagency Working Group
TWG	Technical Working Group – A small group of people overseeing this development effort.
UML	Unified Modeling Language – defines modeling tools that will be used as an aid in extracting requirements and system design.
USACE	U. S. Army Corps of Engineers
USGS/WRD	U. S. Geological Survey, Water Resources Division
XML	Extensible Markup Language – More flexible than HTML, XML can be used to describe any kind of data.